

D i p l o m a r b e i t

zum Thema

“Semantische Dienste in heterogenen Knowledge-Management- und
eLearning-Umgebungen“

eingereicht an der	Wirtschafts- und Sozialwissenschaftliche Fakultät der Universität Rostock
vorgelegt von	Ralph von Petersdorff-Campen
Matrikel-Nr.:	95201417
Diplomstudiengang:	Wirtschaftsinformatik
Bearbeitungszeitraum:	6 Monate
Gutachter:	Prof. Dr. Andreas Heuer, Universität Rostock Prof. Dr. Peter Forbrig, Universität Rostock
Betreuer:	Dipl.-Inf. Gunnar Weber, Universität Rostock Dipl.-Inf. Guntram Flach, ZGDV e.V. Rostock Dipl.-Inf. Matthias Rust, ZGDV e.V. Rostock Dipl.-Inf. Stefan Audersch, ZGDV e.V. Rostock
Lehrstuhl:	Fakultät der Informatik und Elektrotechnik, Institut für Informatik

Rostock, 21. Oktober 2004

Zusammenfassung

Die Verknüpfung von Kursen in eLearning-Systemen mit den Inhalten aus anderen Wissensquellen stellt eine große Herausforderung dar. Verschiedene heterogene Schnittstellen erschweren eine Nutzung von automatischen Verfahren für diese Föderation.

In dieser Arbeit wird ein Ansatz entwickelt, durch welchen mittels semantischer Dienste eine Verknüpfung von Inhalten aus eLearning- und Knowledge-Management-Umgebungen erreicht werden soll. Als Dienste werden in der Arbeit Webservices verwendet. Diese Webservices sollen durch Semantic Web-Technologien um Semantik erweitert werden.

Unter Zuhilfenahme von Metadaten und Ontologien wurde ein Algorithmus entwickelt, welcher es ermöglicht, in verschiedenen eLearning-Systemen nach Kursteilen zu suchen. Diese werden durch den Algorithmus zu einem neuen Kurs föderiert. Für die nötigen Transformationen der Anfragen wurde ein bestehender Gatewayansatz modifiziert.

Die in der Arbeit gewonnenen theoretischen Erkenntnisse wurden in einem Prototyp umgesetzt. Das Hauptaugenmerk lag dabei in der Implementation des Suchalgorithmus und der Einbindung der Wissensdaten aus den Knowledge-Management-Systemen.

Abstract

The connection between courses containing different knowledge sources in eLearning systems are still quite a challenge. Various heterogeneous interfaces complicate the use of automatic methods for federation.

This paper develops an approach which connects content from both eLearning systems and knowledge management systems by means of semantic services. The approach meets this goal by semantically enhanced Web Services using Semantic Web technologies.

Meta data as well as ontologies serve as a base for an algorithm which is capable of searching for appropriate course elements in various eLearning systems. The transformations necessary are done by a modified gateway approach.

The theoretical insights gained in this paper are implemented in a prototype. Focus of this paper was the implementation of the search algorithm and the integration of data from knowledge management systems.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Anwendungsszenario	2
1.3. Genutzte Vorarbeiten	4
1.4. Anforderungen	5
1.5. Aufbau der Arbeit	6
2. Grundlagen und Begriffe	8
2.1. E-Learning	8
2.1.1. SCORM	8
2.1.2. Learning Object Repository	9
2.1.3. Client/Server vs. P2P	10
2.1.4. Mobiles Lernen	11
2.1.5. Didaktik im eLearning-Kontext	11
2.2. Wissensmanagementsysteme	14
2.3. Semantic Web	14
2.3.1. Sprachen zur Wissenspräsentation	16
2.3.2. Ontologie	16
2.4. Service-orientierte Architektur	17
2.4.1. Web Services	19
3. Ansätze für die Föderation von LMS und KMS	22
3.1. Allgemeines	22
3.2. BELLE	23
3.2.1. Peer Review	24
3.2.2. Content Repurposing	24
3.3. Merlot	25
3.3.1. Peer Reviewing	25
3.4. EdnA Online	26
3.4.1. Parameter für die Anfrage	27
3.4.2. Ergebnis der Anfrage	28
3.5. SMETE	29
3.5.1. Web Service-Schnittstelle	30
3.6. Edutella	32
3.6.1. Query Service	33
3.6.2. Mapping Service	35
3.7. Vergleich der Systeme und Fazit	36

4. Lösungsansatz	38
4.1. Architektur	39
4.2. Ontologien und Metadaten	41
4.2.1. Ontologie	42
4.2.2. Metadaten der Lernobjekte	47
4.3. Didaktische Konzepte	51
4.4. WIESELfederation	52
4.5. Transformation Engine	54
4.5.1. Gateway	54
4.5.2. Semantische Transformation	58
4.6. Federation Engine	61
4.6.1. Aspekte für die Föderation	62
4.6.2. Motivation des Suchansatzes	62
4.6.3. Wahl eines Suchansatzes	63
4.6.4. Suche nach einer Heuristik	63
4.6.5. Der Suchalgorithmus	65
4.6.6. Course Enrichment Engine	67
4.6.7. Erstellen eines Gesamtkurses	68
4.7. WIESELMobile	69
4.8. Zusammenfassung und Fazit	71
5. Prototyp und Implementation	74
5.1. Verwendete Technologien	74
5.2. Erweiterungen des LMS	75
5.3. Föderation von Inhalten aus LOR's	77
5.3.1. Verwendung des LOR-Index	77
5.3.2. Verwendung der Fachontologie	78
5.3.3. Gateway zur Anfrage an die LOR's	80
5.3.4. Bestimmung der Relevanz	81
5.3.5. Sortierung der gefundenen SCO's	82
5.4. Einfügen von Inhalten aus KMSs	83
5.5. Erzeugung von Kohärenz	84
5.6. Erstellen der Kursdatei	85
5.7. Zusammenfassung und Fazit	85
6. Zusammenfassung und Ausblick	87
6.1. Zusammenfassung	87
6.2. Ausblick	89
A. Anhang	90
A.1. SMETE Sucheergebnis	90
A.2. Suchalgorithmen	92

Abkürzungsverzeichnis

<i>ADL</i>	Advanced Distributed Learning
<i>AICC</i>	Aviation Industry CBT Committee
<i>API</i>	Application Programming Interface
<i>CAM</i>	Content Aggregation Model
<i>CBT</i>	Computer Based Training
<i>DRM</i>	Digital Right Management
<i>ECDM</i>	Edutella Common Data Model
<i>HTML</i>	Hypertext Markup Language
<i>IEEE</i>	Institute of Electrical and Electronics Engineers, Inc.
<i>IMS</i>	Instructional Management Systems
<i>JXTA</i>	juxtapose
<i>LMS</i>	Lernmanagementsystem
<i>LOM</i>	Learning Object Metadata
<i>LOR</i>	Learning Object Repository
<i>LRMMS</i>	Learning Resource Metadata Management System
<i>OAIPMH</i>	Open Archives Protocol for Metadata Harvesting
<i>P2P</i>	Peer-to-peer
<i>RDF</i>	Resource Description Framework
<i>RDF – QEL – i</i> ...	RDF Query Exchange Language
<i>RTE</i>	Run-Time Environment
<i>SCO</i>	Sharable Content Object
<i>SCORM</i>	Sharable Content Object Reference Model
<i>UMTS</i>	Universal Mobile Telecommunications System
<i>WLAN</i>	Wireless Local Area Network
<i>WMS</i>	Wissensmanagementsystem
<i>XHTML</i>	Extensible Hypertext Markup Language
<i>XSL</i>	Extensible Stylesheet Language
<i>XSLT</i>	Extensible Syslesheet Language Transformation

1. Einleitung

1.1. Motivation

Die momentan eingesetzten Lernmanagementsysteme (LMS) sind zum größten Teil abgeschlossene Systeme in denen komplette Kurse angeboten werden. Dies hat zur Folge, dass für unterschiedliche Nutzer der gleiche Kurs angeboten wird. Eine Erzeugung von individuellen Kursen für einen Nutzer, könnte Voraussetzungen und Neigungen des Nutzers mit einbeziehen. Der Nutzer würde hierdurch nur für in relevante Informationen erhalten. Erste Versuche zur Individualisierung von Kursen werden durch bestehende LMS gegeben, in dem Anpassungen des Layouts (Schriftgröße, Farben, ...) der Kurse durch den Nutzer möglich sind.

Wegen einer nicht vollständigen Standardisierung existieren in der Regel sehr heterogene Systeme, wodurch ein Austausch einzelner Kursteile zwischen den Systemen nicht oder nur schwer möglich ist. Im Zuge der immer stärkeren Vernetzung ist es aber nötig, mögliche Potentiale der Verknüpfung verschiedener LMS zu erkennen und diese zu nutzen.

Neben Lernmanagementsystemen existieren weitere Systeme, welche Wissen digital aufbereitet enthalten. Zu diesen Systemen zählen die Wissensmanagementsysteme (WMS). Diese haben sich in den letzten Jahren entwickelt, um allgemeingültiges Wissen digital zu speichern und somit auch für andere nutzbar zu sein. Durch eine Verknüpfung dieser Inhalte mit den Inhalten aus LMS kann eine Anreicherung der Kurse innerhalb eines LMS erreicht werden.

Das Grundproblem bei der Nutzung von Kursteilen aus verschiedenen LMS und verschiedenen WMS ist der unterschiedliche Aufbau der Daten. Weiterhin müssen die Daten mit zusätzlichen Informationen versehen werden, um eine Suche nach den gewünschten Informationen zu ermöglichen. Um dies zu erreichen, werden heutzutage in der Praxis XML-Metadaten verwendet.

Eine Kopplung verschiedener LMS und WMS kann durch die Nutzung von Service-orientierten Architekturen erreicht werden. Diese bieten eine syntaktische Beschreibung der Schnittstellen, über welche auf die Systeme zugegriffen werden kann. Hierdurch entsteht eine lose Kopplung zwischen den Systemen. Eine Anpassung der Schnittstellen ist in den meisten Fällen nicht nötig, wenn innerhalb der Systeme Veränderungen vorgenommen werden. Ein Problem, welches in diesem Zusammenhang nicht ausreichend geklärt ist, ist die Beschreibung der Semantik innerhalb von Service-orientierten Architekturen. Dies führt dazu, dass die Semantik der Daten der einzelnen Systeme jedem

anderen System bekannt sein muss. Erste Ansätze versuchen dieses Problem mittels Techniken des Semantic Web zu lösen.

Neben dieser rein inhaltlichen Kopplung verschiedener Kursteile kann auch eine geräteabhängige Auswahl und Verknüpfung von Kursen erfolgen. Hierdurch kann dem Trend Rechnung getragen werden, Lerninhalte auch mobil nutzen zu können. Neue mobile Endgeräte haben durch UMTS und WLAN die Möglichkeit, immer größere Dateneinheiten zu erhalten, wodurch auch in mobilen Szenarien die Durchführung von Kursen möglich geworden ist. Ein weiterer Punkt in diesem Zusammenhang ist die Nutzung von kontextabhängigen Kursinhalten. Die Auswahl der Kursinhalte kann aufgrund der Position und Umgebung des Nutzers erfolgen. Diese Inhalte werden dem Nutzer dann als Kurs bereitgestellt.

Im Rahmen des WIESEL-Projektes, welches von der Anova GmbH in Zusammenarbeit mit dem ZGDV (Zentrum für Graphische Datenverarbeitung) Rostock entwickelt wird, soll untersucht werden, welche Möglichkeiten zur Verbindung von Inhalten aus verschiedenen LMS und KMS bestehen. Hierbei sollen die Probleme der Heterogenität zwischen den Systemen betrachtet werden. Der Umgang mit den bestehenden Heterogenitäten soll durch die Verwendung von semantischen Diensten erfolgen. Weiterhin sollen Untersuchungen durchgeführt werden, inwieweit es möglich ist, Kohärenz innerhalb dieser förderierten Kurse zu erzeugen.

1.2. Anwendungsszenario

Innerhalb des Anwendungsszenario soll die Umsetzung einer Vorlesung über Zellbiologie als Online-Kurs entstehen. Die Vorlesung wird von Prof. Dr. Weiss, welcher im Institut für Biologie an der Universität Rostock tätig ist, gehalten. Aufbauend zur Vorlesung soll den Studenten die Möglichkeit gegeben werden, einen Online-Kurs zum Vertiefen des Wissens aus der Vorlesung zu besuchen. Weiterhin sollen zusätzliche Inhalte aus anderen Wissensquellen innerhalb des Kurses genutzt werden. Als Wissensquellen stehen eine Videodatenbank, eine Definitionsdatenbank und Powerpoint-Präsentation zur Verfügung.

Die Umsetzung dieses Anwendungsszenarios soll innerhalb des Lernmanagementsystems der Firma Anova GmbH erfolgen. Dazu wurde eine Architektur entwickelt, die die Anforderungen des Anwendungsszenarios erfüllt.

Innerhalb der Architektur werden neben der Funktionalität eines LMS auch folgende Funktionen unterstützt:

- Bereitstellung der Kursteile in einem Learning Object Repository (LOR)
- Erstellung von Kursen bzw. Kursteilen mittels spezieller Tools und
- Darstellung von Inhalten aus Wissensmanagementsystemen (WMS).

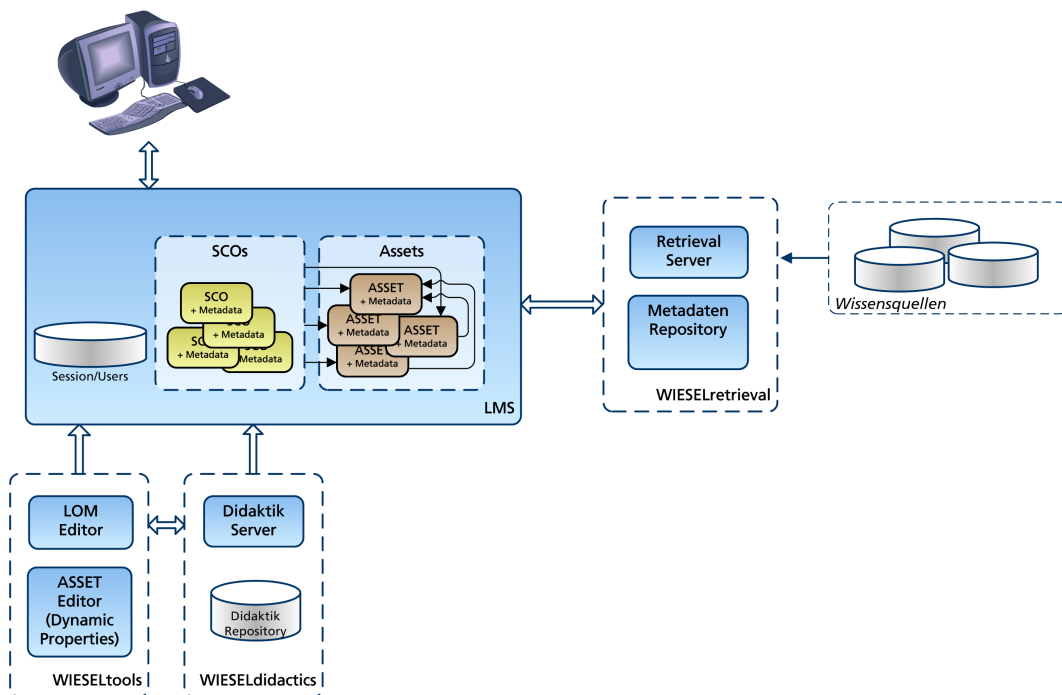


Abbildung 1: Das WIESELFramework ohne Erweiterungen aus der Arbeit

Diese obengenannten Funktionen sind in der Abbildung 1 abgebildet. In der Abbildung 1 sind die entsprechenden Module dargestellt. Diese werden nachfolgend näher erläutert.

Das Modul WIESELTOOLS stellt die Softwarelösungen bereit, welche dem Autor Möglichkeiten bieten, Kurse bzw. Kursteile zu erstellen. Die Programme zum Erstellen von Kursteilen bieten Hilfestellungen, um diese Inhalte nach didaktischen Modellen zu generieren. Die zur Verfügung stehenden didaktischen Modelle sind im Didaktik-Server des Modules WIESELDidactics gespeichert. Zusätzlich existieren Programme mit denen Metadaten zu Kursinhalten angegeben werden können und Kursteile zu Kursen zusammengefasst werden.

Im Modul WIESELRETRIEVAL existiert ein Repository, welches die verschiedenen Wissensdaten enthält. Weiterhin existiert eine Web Service-Schnittstelle, über welche Suchanfragen gestellt werden können. Um eine Nutzung der Daten innerhalb des Lernmanagementsystems zu ermöglichen, wurde ein Tag `<dynamiccontent>` definiert, welches durch das LMS aufgelöst wird. Das LMS sendet die, durch das Tag gelieferten, Daten an das Repository. Das Ergebnis ist ein XML-Dokument, welches mittels Stylesheets an das Layout des Kurses angepasst wird. Dieses Modul spielt eine große Rolle in der Arbeit. Daher wird es im nächsten Abschnitt noch einmal genauer betrachtet.

1.3. Genutzte Vorarbeiten

In diesem Abschnitt soll auf zwei Softwarelösungen, welche im ZGDV Rostock entwickelt wurden eingegangen werden, da sie innerhalb der Arbeit verwendet werden.

Zum einen handelt es sich um das, im vorherigen Abschnitt beschriebene, Modul WIESELRETRIEVAL. Zum anderen geht es um einen Gatewayansatz, welcher innerhalb einer Diplomarbeit entstanden ist.

Wie oben beschrieben dient das Modul WIESELRETRIEVAL der Bereitstellung von Wissen aus einem Wissensrepository. Das Repository enthält Daten (Videos, Definitionen, ...) aus dem Bereich Zellbiologie. Weiterhin sind verschiedene Ontologien im Repository gespeichert. Der Zugriff auf die Daten erfolgt mittels einer Suchanfrage an den zur Verfügung stehenden Web Service. Mittels des Schlüssels `applicationContext` kann auf die verschiedenen Datenquellen zugegriffen werden. Zur Verfügung stehen

- `repositoryInfo`,
- `ontoResmo` und
- `fileInfo`.

Über den `applicationContext repositoryInfo` können alle verfügbaren Datenquellen erfragt werden. Als Ergebnis werden neben der ID der Datenquelle auch die Parameter für die Anfrage an die Datenquelle übermittelt. Der `applicationContext fileInfo` liefert die im angegebenen Dateisystem vorhandenen Daten. Ein Punkt, welcher für die Arbeit wichtig ist, ist der `applicationContext ontoResmo`. Dieser `applicationContext` ermöglicht den Zugriff auf die im Repository enthalten Ontologien. Es stehen mehrere Möglichkeiten für die Nutzung der Beziehungen in der Ontologie zur Verfügung, z.

B. Synonyme, Subklassen, Superklassen, Die genaue Verwendung dieser Schnittstelle zu den Ontologien wird im Abschnitt 4.2.1 des Kapitels 4 erläutert.

Der Gatewayansatz wurde in der Diplomarbeit von Matthias Schultz [Sch03] erarbeitet. Das Anwendungsszenario, in welchem dieser Gatewayansatz genutzt wird, ist ein Bürgerkontaktsystem. Mittels dieses Kontaktsystems wird es den Bürgern ermöglicht, Dienste von Behörden online in Anspruch zu nehmen. Hierzu kann der Bürger Online-Formulare ausfüllen. Die Daten der Formulare werden in ein Format eines bestimmten Fachverfahrens gebracht. Ein Fachverfahren ist in diesem Zusammenhang ein System, welches zur Bearbeitung der Daten eines Online-Dienste genutzt wird. Durch das System wird die Darstellung der eingegebenen Daten beschrieben. Im Bereich des eGovernments gibt es keine Standardisierung der Fachverfahren, daher existieren verschiedene Verfahren mit unterschiedlichen Schnittstellen und Datenpräsentationen. Der in der Diplomarbeit von Matthias Schulz [Sch03] entwickelte Ansatz ermöglicht eine automatische Transformation der Schnittstellen und Datenformate der verschiedenen heterogenen Fachverfahren.

1.4. Anforderungen

Diese Arbeit beschäftigt sich mit der Umsetzung der Föderation von Lerninhalten aus heterogenen LMS und WMS und bindet die gewonnenen Erkenntnisse in das Framework ein.

Allgemeine Anforderungen an die Arbeit sind:

- **lose Kopplung der verschiedenen Module**
Durch eine lose Kopplung der Module soll erreicht werden, dass die einzelnen Module ausgetauscht werden können.
- **Unabhängigkeit**
Die erstellten Komponenten sollen unabhängig von dem darunterliegenden Betriebssystem sein. Weiterhin sollen die Komponenten auch unabhängig von dem darüberliegenden Lernmanagementsystem sein.
- **Konsistenz**
Die Daten, die für die Kommunikation zwischen den verschiedenen Systemen verwendet werden, dürfen in keinem Zwischenschritt verfälscht werden. Hierfür ist es nötig, syntaktische aber auch semantische Betrachtungen vorzunehmen.

- **Erweiterbarkeit**

Der Aufbau des Systems soll modular gestaltet sein, um spätere Erweiterungen integrieren zu können.

Neben diesen allgemeinen Anforderungen ergeben sich spezielle Ansprüche an die Arbeit:

- **Einbindung semantischer Daten**

Für die Verknüpfung von Inhalten aus verschiedener Systeme besteht die Notwendigkeit, semantische Informationen über die Verwendung der enthaltenen Daten bereitzustellen. Diese Semantikbeschreibung soll über die Schnittstellenbeschreibungen der Systeme an die angegliederten Module übermittelt werden. Weiterhin müssen Verfahren gefunden werden, wie semantische Dienste genutzt werden können, um Anfragen an die verschiedenen Systeme zu erweitern. Dies ist unter dem Gesichtspunkt der Verbesserung des Ergebnisvektors zu untersuchen.

- **Erweiterung von eLearning-Inhalten durch Knowledge-Management-Inhalte**

Neben der Betrachtung von Inhalten aus eLearning-Umgebungen soll untersucht werden, wie diese Inhalte durch andere Wissensquellen innerhalb der Unternehmen erweitert werden können.

- **Verwendung von Daten über das Endgerät**

Die Daten über das verwendete Endgerät des Nutzers soll genutzt werden, um die förderierten Kurse an das Gerät anzupassen. Hiermit soll eine Nutzung durch mobile Endgeräte ermöglicht werden.

- **Berücksichtigung didaktischer Konzepte**

Bei der Föderation von Inhalten aus verschiedenen Wissensquellen ist es nötig, Konzepte zu berücksichtigen, die die Anordnung der Kursobjekte unter dem Gesichtspunkt der Didaktik ermöglichen.

- **Nutzung von Technologien zur Kohärenzzeugung**

In unterschiedlichen Systemen werden unterschiedlich formatierte Inhalte präsentiert. Daher ist es nötig, dass Kohärenz durch geeignete Verfahren hergestellt wird.

1.5. Aufbau der Arbeit

Die Arbeit gliedert sich in mehrere Bereiche, die im nachfolgenden kurz klassifiziert werden sollen. Als Erstes werden Definitionen wichtiger Begriffe gegeben.

Nachfolgend wird eine Untersuchung bestehender Ansätze vorgenommen. Ausgehend von diesen Ansätzen wird ein eigener Lösungsansatz entwickelt, welcher in einem Prototypen umgesetzt wird. Abschließend wird eine Zusammenfassung und ein Ausblick gegeben.

Begriffserklärungen

In diesem Abschnitt werden die verwendeten Begriffe aus dem Bereich eLearning, Wissensmanagement und Semantic Web erklärt.

Bestehende Ansätze

Der Abschnitt beschäftigt sich mit Ansätzen, die eine Föderation von eLearning-Inhalten anbieten. Am Ende dieses Abschnittes wird ein Vergleich der betrachteten Systeme vorgenommen.

Lösungsansatz

Dieser Abschnitt stellt den entwickelten Ansatz dar. Hierbei werden die Verwendung der semantischen Dienste beschrieben. Weiterhin wird beschrieben, wie diese Dienste genutzt werden, um eine Verknüpfung von Inhalten aus eLearning- und Knowledge-Management-Umgebungen zu erreichen. Neben diesen Techniken werden Ansätze aufgezeigt, wie mittels didaktischer Konzepte und Verfahren zur Herstellung der Konsistenz, diese Verknüpfung verbessert wird.

Prototyp

Die Umsetzungen des Lösungsansatzes durch den Prototypen wird in diesem Abschnitt erläutert.

Zusammenfassung und Ausblick

Abschließend wird in diesem Kapitel eine Zusammenfassung der Arbeiten in dieser Diplomarbeit aufgezeigt. Hier wird untersucht, wie die Anforderungen durch den Lösungsansatz und dem daraus entstandenen Prototypen umgesetzt wurden. Im Ausblick werden Ansätze für weiterführende Arbeiten gegeben. Hierbei werden die im Lösungsansatz gefundenen möglichen Verbesserungen angesprochen.

2. Grundlagen und Begriffe

2.1. E-Learning

2.1.1. SCORM

Das Shareable Content Object Reference Model (SCORM) [ADL04] ist kein Standard, da das Model eine Spezifikation bereitstellt, welche aus Teilen anderer eLearning-Spezifikationen besteht. Für die SCORM-Spezifikation werden Teile aus den Spezifikationen der IMS, AICC und IEEE verwendet. Ziel der Advanced Distributed Learning (ADL)-Gruppe, welche diese Spezifikation entwickelt, ist es, Entwicklern die Möglichkeit zu geben, wiederverwendbare Kurse zu erstellen.

Diese Wiederverwendbarkeit kann auf drei Ebenen erfolgen, die dem festgelegten Aufbau von SCORM-Kursen entsprechen. Ein SCORM-Kurs kann aus nur einer SCO bestehen oder aus mehreren SCO's zusammengesetzt worden sein. Ein SCO besteht aus verschiedenen Assets. Jedes Asset kann wiederum aus mehreren Assets bestehen. Ein Asset ist hierbei immer eine Komponente, die in einem Webbrowser dargestellt werden kann. Ein Beispiel für ein zusammengesetztes Asset ist eine Webseite, die sowohl aus Text, als auch Bildern und anderen Elementen besteht. Der Unterschied zwischen SCO und Asset besteht darin, dass ein SCO neben den reinen Inhaltsdaten auch Informationen enthält, die für die Kommunikation mit dem LMS erforderlich sind. Durch diese zusätzlichen Informationen ist es möglich, SCO's zwischen verschiedenen LMS auszutauschen.

Die einzelnen Komponenten eines SCORM-Kurses werden durch Metadaten beschrieben, um die Wiederverwendbarkeit zu verbessern. Innerhalb der Metadaten wird auch beschrieben, wie die verschiedenen SCO's in dem Kurs angeordnet sind. Durch diese Definition der Reihenfolge der SCO's können einzelne SCO's zu komplexeren SCO's verbunden werden.

Innerhalb der SCORM-Spezifikation werden drei Komponenten definiert, welche für die Beschreibung und Nutzung eines SCORM-Kurses nötig sind. Diese sind:

- das **Content Aggregation Model (CAM)**,
- die **Run Time Environment (RTE)** und
- das **Sequencing and Navigation (SN) model**.

Das **Content Aggregation Model** definiert, wie Lernobjekte beschrieben, vereint und in eine Reihenfolge gebracht werden müssen. Es beschreibt, wie die Lernobjekte verpackt werden müssen, um einen Austausch zwischen Systemen zu ermöglichen.

Die **Run Time Environment** beschreibt die Kommunikation zwischen Kursen und dem Lernmanagementsystem. Weiterhin werden die Vorgaben beschrieben, die ein Lernmanagementsystem umsetzen muss.

Das **Sequencing and Navigation**-Model beschreibt, wie der Inhalt der Kurse durch Aktionen des Lernenden bzw. des Systems in eine bestimmte Reihenfolge gebracht wird. Zusätzlich werden beschrieben, wie das Lernmanagementsystem die Reihungsregeln interpretieren muss. Dieses Model existiert erst ab der Version SCORM 2004.

2.1.2. Learning Object Repository

Ein Learning Object Repository (LOR) ist ein System, welches Daten aus dem Bereich eLearning bereitstellt und verwaltet. Die in einem LOR enthaltenen Daten werden auch als Learning Objects bezeichnet. In LOR's, die sich an die SCORM-Spezifikation halten, sind SCO's als Learning Objects gespeichert.

Wichtig für ein LOR ist die Bereitstellung einer Schnittstelle für das Durchsuchen des Inhaltes. Für das Durchsuchen des Inhaltes ist es nötig, dass die enthaltenen Learning Objects durch Metadaten beschrieben sind.

Ein Punkt, der bei der Generierung und Bereitstellung von Learning Objects eine Rolle spielt, ist die Größe des Learning Objects. Ein Learning Object sollte nach Möglichkeit nur ein definiertes Lernziel erklären. Je geringer die Größe der Learning Objects ist, desto besser lassen sich aus diesen Lernobjekten Kurse zusammenstellen. Es muss aber so umfangreich sein, dass die nachfolgenden Eigenschaften nicht verletzt werden.

Ein Lernobjekt sollte folgende Eigenschaften erfüllen [Ham03]:

- Konsistenz - geschlossen, lückenlos und widerspruchsfrei
- Eigenständigkeit - allein stehend verständlich
- Austauschbarkeit - ersetzbar durch anderes LO
- Aggregierbarkeit - kombinierbar mit anderen LO und
- Portabilität - übertragbar in andere Kontexte.

Bei der Aggregation von Lernobjekten müssen auch folgende Sichten berücksichtigt werden [Ham03]:

- thematische Sicht - geschlossenes Thema pro Modul
- didaktische Sicht - Berücksichtigung der Lernsituation, Lernziele und Umgebungsbedingungen des Nutzers
- sprachliche Sicht - gleiche Fachbegriffe und gleiche semantische Dichte
- soziokulturelle Sicht - Kontextwissen, z. B. Maßangaben, Feiertage, ...
- gestalterische Sicht - graphische Gestaltung und
- technische Sicht - Hard- und Softwareausstattung der Zielgruppe.

Durch diese Definition von Sichten entstehen Cluster mit Learning Objects die gut zu einem Kurs zusammengefasst werden können, da keine Heterogenitäten im jeweiligen Bereich der Sicht auftreten.

2.1.3. Client/Server vs. P2P

Für den Austausch von eLearning-Daten werden sowohl Client/Server-Architekturen genutzt, als auch P2P-Netze. Überwiegend werden im eLearning-Bereich Client/Server-Architekturen eingesetzt. Hier sollen die Unterschiede zwischen den beiden Varianten des Datenaustausches dargestellt werden.

Der Austausch von Daten aus unterschiedlichen Systemen erfordert immer eine Kommunikation zwischen den anbietenden und den nachfragenden Parteien. Für diese Kommunikation bieten sich zwei grundsätzliche Methoden an.

Bei der Nutzung von Servern wird die Kommunikation der anfragenden Partei minimiert, da er nur einen Anlaufpunkt (den Server) hat. Dem steht der höhere administrative Aufwand gegenüber, welcher durch die Verwaltung der Daten, welche auf dem Server gespeichert sind, entsteht. Die Daten auf dem Server müssen durch regelmäßige Updates auf den neusten Stand gebracht werden. Sind die Updateabstände zu groß, kommt es zu Inkonsistenzen zwischen den Daten des Servers und den Daten des Autors.

Die andere Methode ist die Nutzung des P2P-Konzepts. Ein wesentlicher Vorteil ist die direkte Kommunikation zwischen den Partnern, wodurch es zu keinen Inkonsistenzen kommen kann, da nur das System, welches die Daten anbietet, die entsprechenden Metadaten bereitstellt. Auf der anderen Seite muss aber jeder Partner alle anderen Partner kennen, um effektiv und effizient die

benötigten Daten anfragen zu können. Da dies ab einer bestimmten Größe des P2P-Netzes nicht mehr möglich ist, wurden neue Techniken entwickelt, welche ein Weiterreichen der Anfragen durch das Netz bewirken. In den meisten Fällen erfolgt eine Speicherung der Partner in einer Datei und bei jeder Anfrage werden gleichzeitig diese Daten zwischen den Partnern ausgetauscht.

Neben diesen reinen Formen gibt es auch Mischformen. In diesen Mischformen verwalten die Server die verschiedenen Clients. Der Austausch der Daten erfolgt nicht über den Server, sondern zwischen den Clients direkt. Eine Erweiterung dieses Ansatzes ist die Speicherung der Metadaten über die Inhalte der einzelnen Clients auf dem Server. Hierdurch müssen die Anfragen nicht vom Server an die Clients weitergereicht werden.

2.1.4. Mobiles Lernen

Bis vor kurzem war das mobile Lernen beschränkt auf die Nutzung von Offline-Kursen. Dies hat den Nachteil, dass Kurse auf dem mobilen Endgerät unabhängig von einem LMS durchgeführt werden und damit erbrachte Leistungen nur schwer digital erfasst werden können.

Die erste Schwierigkeit, die beim Darstellen von Kursinhalten eines LMS auf mobilen Endgeräten auftritt, ist die Anpassung der HTML-Seiten, aus welchen die meisten Kurse bestehen, für die Geräte. Durch die Verwendung von XHTML und XSLT kann diese Anpassung einfach erfolgen. Hierdurch können sehr große Webseiten geteilt werden, Bilder skaliert und nicht anzeigbare Elemente entfernt werden.

Ein Problem bei der Anzeige von SCORM-Kursen liegt bei den Mindestanforderungen an den Browser. Die Anforderungen sind die Unterstützung von Frames, Javascript und Java durch den Browser. Auf den meisten mobilen Systemen stehen diese Voraussetzungen nicht zur Verfügung, wodurch es nötig ist, andere Möglichkeiten der Präsentation zu finden.

2.1.5. Didaktik im eLearning-Kontext

Ein bestehender Ansatz für die Nutzung von didaktischen Konzepten in eLearning-Kursen ist in der Spezifikation des IMS mit IMS-LD [IMS] gegeben. Eine weitere DIN-Spezifikation ist die PAS 1032 [din]. Sie ist keine Norm, kann aber als Ausgangspunkt für eine Normung genutzt werden. Diese Spezifikation besteht aus zwei Teilen. Der erste Teil bietet ein "Referenzmodell für Qualitätsmanagement und Qualitätssicherung - Planung, Entwicklung, Durchführung

und Evaluation von Bildungsprozessen und Bildungsangeboten“ [din]. Dieser Teil ist für die Arbeit nicht relevant. Im zweiten Teil wird ein “Didaktisches Objektmodell - Modellierung und Beschreibung didaktischer Szenarien“ [din] bereitgestellt. Das Beschreibungsmodell wird in der folgenden Abbildung dargestellt.

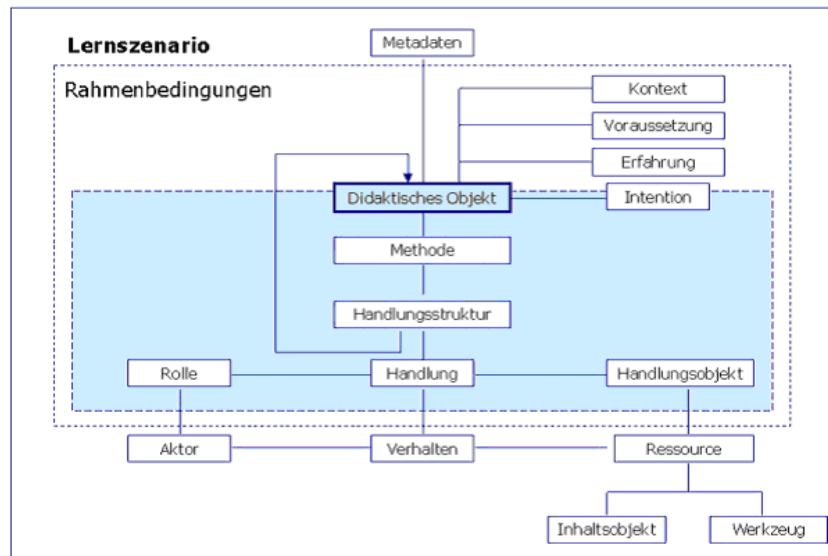


Abbildung 2: Beschreibungsmodell nach der DIN-Spezifikation PAS 1032-2[din]

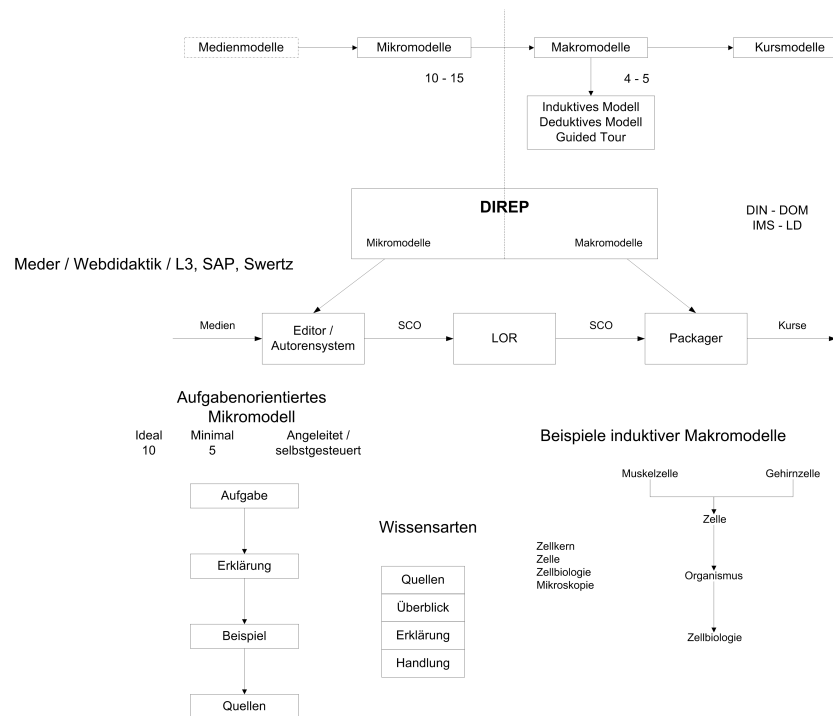


Abbildung 3: Didaktische Betrachtungen in der Arbeit

Wie in Abbildung 3 gezeigt wird, können innerhalb des eLearnings didaktische Untersuchungen auf vier verschiedenen Ebenen erfolgen.

Die erste Ebene betrachtet die Aufbereitung von atomaren Modulen (Bilder, Videos, Texte, Grafiken, ...). Dies sind die sogenannten **Medienmodelle**.

Im zweiten Punkt geht es um die Erzeugung von SCO's, welche einem bestimmten **Mikromodell** entsprechen.

Der dritte Schritt sind die **Makromodelle**. Diese werden genutzt, um die einzelnen SCO's zu Kursen zusammenzufügen.

Zu der vierten Klasse von Modellen, den sogenannten **Kursmodellen**, zählen Modelle, die für die Verbindung von verschiedenen Kursen zuständig sind. Bei diesen Ansätzen werden die Relationen zwischen zusammengehörenden Kursen betrachtet.

Die hier genannten Klassen von Modellen sind ein Teilaspekt der Webdidaktik, welche unter [web] und bei [Swe04] beschrieben ist.

2.2. Wissensmanagementsysteme

Im Bereich der Föderation von Inhalten aus Wissensmanagementsystemen (Knowledge Management System) KMS und LMS ist ein Wissensmanagementsystem ein System, welches Wissen speichert und verwaltet. Ein wichtiger Punkt bei der Generierung von KMS ist die Bereitstellung von Schnittstellen, um in den Inhalten suchen zu können.

Das in diesen Systemen beschriebene Wissen kann in beliebiger Form vorliegen. Die Suche nach Inhalten erfordert eine Beschreibung der enthaltenen Inhalte durch Metadaten. Durch diese Eigenschaften ähneln sie stark den oben beschriebenen LOR's.

Aufgrund der Schnittstelle für die Suche und die Beschreibung durch Metadaten eignen sich diese Systeme gut für die Verbindung mit LOR's, um das in den LOR's dargebotene Lernwissen zu erweitern.

Bekannte Systeme sind die Entwicklung von digitalen Bibliotheken, welche in die Lernumgebung integriert werden. Ein Beispiel hierfür ist LEBONED, welches beschrieben wird in [OMR03].

2.3. Semantic Web

“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.“ [BLHL01]

Das Semantic Web kann als nächste Generation des Internets angesehen werden. Nachdem die erste Generation von Webinhalten nur aus statischen Seiten bestand und in der zweiten Generation Webinhalte dynamisch generiert wurden, ist das Semantic Web die Anreicherung der statischen und dynamischen Inhalte um beschreibende Elemente. Diese zusätzlichen Informationen über die Inhalte sollen es ermöglichen, eine Nutzung durch Maschinen zu erleichtern. Die zusätzlichen Informationen werden mittels RDF angegeben.

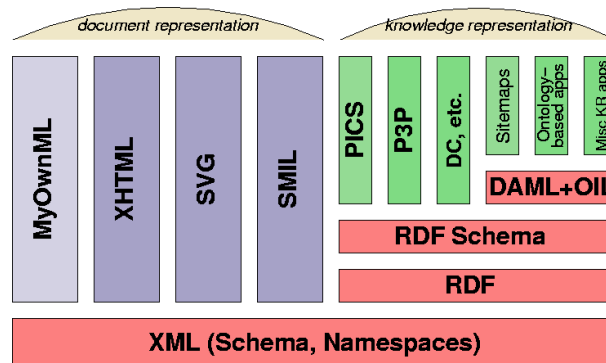


Abbildung 4: Dokument- und Wissenspräsentationssprachen fürs Web []

Die Abbildung 4 zeigt die beiden Sprachbereiche des Semantic Web auf. Die Sprachen für die Dokumentenpräsentation dienen der Darstellung der Inhalte im Semantic Web. Die Sprachen für die Wissenspräsentation beschreiben zusätzliche Daten (Metadaten), um die Inhalte mit einer Semantik anzureichern.

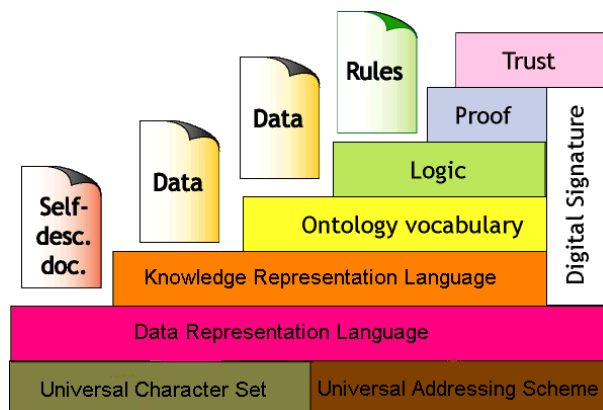


Abbildung 5: Schichten des Semantic Web []

Auf zwei der Schichten in Abbildung 5 wird in den nachfolgenden Unterabschnitten eingegangen. Diese beiden Schichten sind die Sprachen, welche für die Wissenspräsentation genutzt werden und die Ontologieschicht. Die unterhalb dieser Schichten liegenden Schichten dienen der Repräsentation der Inhalte.

Weitere Informationen zu diesem Begriff sind bei [Wor01a] und [BLHL01]

2.3.1. Sprachen zur Wissenspräsentation

Innerhalb dieses Abschnittes sollen verschiedene Sprachen vorgestellt werden, welche für die Beschreibung von Metadaten geeignet sind. Die vorgestellten Sprachen sind jeweils Weiterentwicklungen der vorher dargestellten Sprache. Daher werden nur die vorgenommenen Erweiterungen dargelegt.

Das Resource Description Framework (RDF) stellt zusätzliche Informationen über eine bestimmte URI bereit. Aufgebaut ist RDF als ein Graphensystem, in welchem jeder Teilbaum aus dem Tripel Subjekt, Prädikat und Objekt besteht. Das Subjekt entspricht dem zu beschreibenden Element, das Prädikat ist die Art der Beschreibung und das Objekt ist der Wert der Beschreibung. Die Spezifikation und andere Details sind unter [Wor97] zu finden.

Eine erste Erweiterung ist die Definition mittels RDFS. RDFS (Resource Description Framework Schema) bietet zusätzliche Modellierungsprimitive, wie Klassen, Eigenschaften und Einschränkungen. Die Spezifikation ist unter [Wor04d] zu finden.

In dieser Erweiterung fehlen immer noch wichtige Konzepte, daher wurde die deklarative Sprache DAML+OIL entwickelt. Sie erfüllt alle Bedingungen, die nötig sind, um Wissen umfassend zu beschreiben. Informationen über diese Sprache sind unter [HHPS01] zu finden.

Aufbauend auf DAML+OIL wurde vom W3C die Sprache OWL vorgeschlagen. Diese Sprache hat zum Zeitpunkt der Arbeit den Status “W3C Recommendation“. OWL ist eine Überarbeitung der DAML+OIL-Sprachdefinition. Die entsprechenden Informationen sind unter [Wor04c] zu finden.

2.3.2. Ontologie

Die ursprüngliche Definition der Ontologie leitet sich aus dem griechischen Wort **on**, welches für “Seiendes“ steht, ab. Dieser Lehre des Seins steht die Definition der Ontologie in der Informatik gegenüber. “Unter einer Ontologie versteht man in der Informatik, im Bereich Künstliche Intelligenz, ein formal definiertes System von Dingen und/oder Konzepten und Relationen zwischen diesen Dingen. Zusätzlich enthalten Ontologien (zumindest implizit) Regeln.“[ont]

Die in der Arbeit verwendeten Ontologien sollen möglichst vollständige Beschreibungen der reellen Welt sein. Dabei ist jede Ontologie die Abbildung von Begriffen und deren Beziehungen untereinander innerhalb einer bestimmten Wissensdomäne. Weiterhin können in den Ontologien Verknüpfungen zu

anderen Ontologien bestehen. Durch die Verknüpfung aller Ontologien sollte es möglich werden, die gesamte reelle Welt abzubilden.

Eine Möglichkeit der Beschreibung von Ontologien ist die Nutzung von Topic Maps. Eine mögliche Spezifikation ist unter [Mem01] zu finden. Ein Beispiel für die Verwendung von Topic Maps ist das Wissensnetz von KnowledgeDirect [ZGD03].

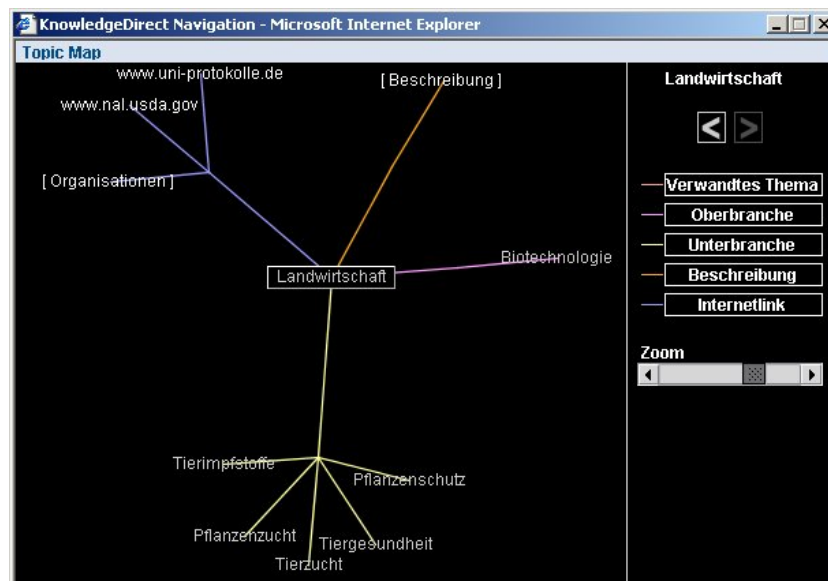


Abbildung 6: Beispiel einer Topic Map im Projekt KnowledgeDirect

Weiterhin kann die Beschreibung der Ontologien durch die im vorangegangenen Abschnitt vorgestellten Sprachen des Semantic Web erfolgen.

Beispiele für Ontologien sind zum Beispiel unter [BA] und [Gen] zu finden. Einige dieser Ontologien wurden in der Arbeit verwendet.

2.4. Service-orientierte Architektur

Eine Service-orientierte Architektur (SOA) dient der Bereitstellung von Dienstleistungen in Computernetzwerken.

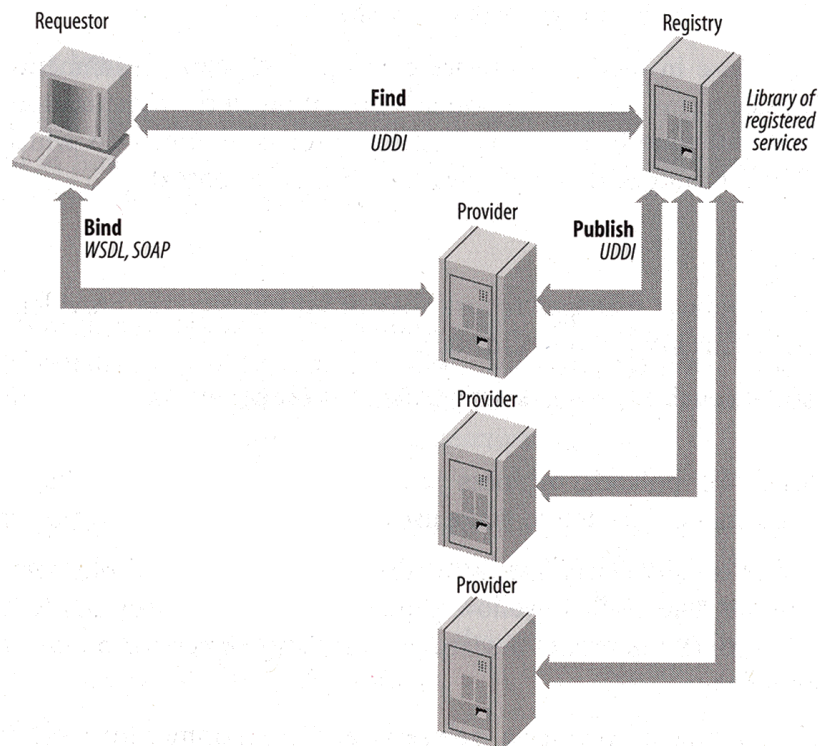


Abbildung 7: Service-orientierten Architektur[CJ02]

In Abbildung 7 ist der Aufbau einer Service-orientierten Architektur dargestellt. Zu sehen ist, wie ein *Requestor* über die *Registry* einen *Provider* findet, welcher einen von ihm benötigten Web Service zur Verfügung stellt. Die Protokolle, welche genutzt werden, um die Daten zwischen den Systemen auszutauschen, werden im nächsten Abschnitt erläutert.

In der nachfolgenden Abbildung werden die Merkmale einer Service-orientierten Architektur dargestellt.

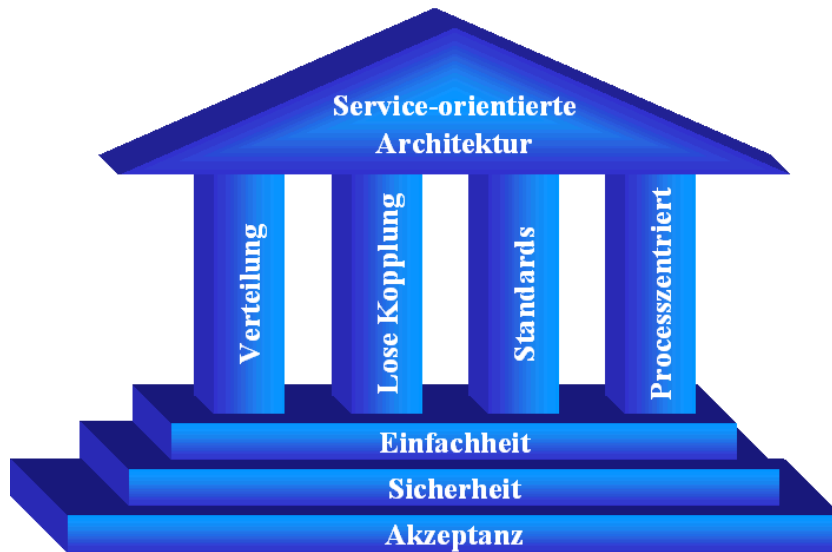


Abbildung 8: Der Tempel der Service-orientierten Architektur[DJ04a]

In der Abbildung 8 können die Eigenschaften einer SOA abgelesen werden. Eine SOA soll einfach, akzeptiert und sicher sein. Ein Merkmal einer SOA ist die Verteilung der Services über das Netzwerk. Ein weiteres Merkmal ist die lose Kopplung zwischen dem Service und dem Nachfrager des Services. Durch Standards kann zwischen verschiedenen Serviceanbietern besser gewechselt werden. Services sollen weiterhin prozessorientiert sein. Das heißt, sie sollen elementare Leistungen anbieten. Um komplexe Leistungen zu erzeugen, wird eine Kette von Services durchlaufen.

Die momentanen Service-orientierten Architekturen bieten nur syntaktische Beschreibungen des Services. Im nächsten Abschnitt wird die Technik der Web Services besprochen, da diese eine weitverbreitete Technik zur syntaktischen Beschreibung von Services ist.

Um SOA für Maschinen nutzbar zu machen, ist es nötig, die syntaktischen Beschreibungen durch semantische Elemente zu erweitern. Hierfür werden Techniken des Semantic Web genutzt.

Eine Beschreibung der Erweiterung von Web Services durch semantische Beschreibungen ist unter [DJ04a], [DJK04a], [DJK04b] und [DJ04b] zu finden.

2.4.1. Web Services

Web Services dienen zur Bereitstellung von Schnittstellen, um über das Internet auf Programme einheitlich zugreifen zu können. Diese Schnittstelle ist

unabhängig von der gewählten Programmiersprache, in der das jeweilige Programm geschrieben ist.

Die wichtigsten Eigenschaften von Web Services sind[CJ02]:

- XML-basiert,
- lose gekoppelt,
- grobkörnig,
- synchron oder asynchron,
- unterstützen Remote Procedure Calls(RPCs) und
- unterstützen Dokumentenaustausch.

Der erste Punkt bedeutet, dass alle Daten, die für die Kommunikation mit Web Services genutzt werden, in XML verfasst sind. Hierdurch wird es möglich Web Services unabhängig von der Plattform und dem Betriebssystem zu nutzen.

Durch die lose Kopplung zwischen Client und dem Web Service ist es möglich, die Programme an die jeweiligen Bedürfnisse anzupassen und zu verbessern, ohne dass sich die Schnittstelle nach außen ändert.

Methoden, welche in objektorientierten Techniken genutzt werden, sind in der Regel zu klein, als dass man mit ihnen einen Service bereitstellen könnte. Daher ist es nötig eine Verkettung von mehreren dieser Methoden zu einer größeren vorzunehmen. Dieser grobkörnige Service kann dann mittels eines Web Service bereitgestellt werden.

Web Services unterstützen sowohl synchrone, als auch asynchrone Anfragen. Bei den synchronen Anfragen wartet der Client nach der Anfrage bis die Antwort ankommt. Im asynchronen Fall kann der Client nach der Anfrage normal weiterarbeiten und nutzt die Antwortdaten erst, wenn sie benötigt werden.

Die Unterstützung von RPC erfolgt durch Bereitstellung der RPC unterhalb der Web Service-Schnittstelle bzw. durch die Übersetzung der Anfragen an den Web Service in Anfragen an die RPCs.

Durch XML ist es weiterhin möglich, komplexe Dokumente zu kapseln, da für den Web Service der Aufbau der übertragenen Daten irrelevant ist.

Die momentan am häufigsten genutzten Web Service-Technologien sind[CJ02]:

- Simple Object Access Protocol (SOAP)[Wor03],
- Web Service Description Language (WSDL)[Wor01b] und

- Universal Description, Discovery, and Integration (UDDI)[OAS01].

SOAP[Wor03] beschreibt Strukturen, wie Dokumente mittels XML über verschiedene Internettechnologien übertragen werden können. Weiterhin beschreibt es wie RPC-Anfragen in ein XML-Format verpackt werden können. Durch SOAP ist es möglich, verschiedene RPC-Systeme miteinander zu koppeln. Hierdurch ist es z. B. möglich, dass .NET-Clients auf EJBs zugreifen können und umgekehrt können Java-Client auf .NET-Applikation zugreifen.

WSDL[Wor01b] dient zur Beschreibung von Web Services. Die Beschreibung der Web Services erfolgt mit Hilfe einer XML-Syntax. Die folgenden Elemente werden benutzt, um einen Web Services zu beschreiben.

types	Definition von Datentypen z.B. mittels XML Schema
message	Definition der Daten, welche für die Kommunikation mit dem Webservice genutzt werden
operation	Definition einer Aktion, die durch den Webservice unterstützt wird
port type	Menge von Operationen, die von einem oder mehreren Endpunkt(en) benutzt wird(werden)
binding	ein konkretes Protokoll und eine konkrete Datenformatspezifikation für einen bestimmten "port type"
port	Definition eines einzelnen Endpunktes, welcher aus einem "binding" und einer Netzadresse besteht
service	Sammlung von möglichen Endpunkten

Tabelle 1: Elemente, die in WSDL beschrieben werden können

Mittels UDDI[OAS01] wird eine Service bereitgestellt, welcher Informationen über Web Services, die bei dem Service registriert sind, bereithält. In dieser Menge von Informationen können die Clients nach den für sie relevanten Services suchen.

3. Ansätze für die Föderation von LMS und KMS

3.1. Allgemeines

Neben den reinen LMS gibt es heutzutage auch mehrere LOR's, die nicht das Vermitteln von Wissen an den Lernenden zum Ziel haben, sondern vielmehr die Lernressourcen für den Lerner bzw. für andere Systeme zur Verfügung stellen.

Diese Systeme sollen in diesem Kapitel untersucht werden. Dabei sollen besonders die Möglichkeiten der Anfrage und Suche in den Systemen dargestellt werden. Einen guten Überblick der Systeme bietet [ND02]. Einige dieser Systeme sollen genauer beleuchtet werden. Wichtig für die Auswahl der Systeme ist die Bereitstellung einer Schnittstelle für die maschinelle Suche und Ansätze für die Föderation verschiedener LOR's.

Ein wichtiges Unterscheidungsmerkmal bei den Systemen ist die Speicherung der Metadaten und der Austausch der Kursinhalte. Zu unterscheiden ist zwischen den serverseitigen bzw. den clientbasierten Systemen.

Bei den serverseitigen Systemen werden die Metadaten der Kurse bzw. der Kursteile auf dem Server gespeichert. Jeder suchende Client stellt eine Anfrage an den Server und erhält von diesem eine Liste von möglichen Kursen. Diese kann er dann direkt von den anbietenden System herunterladen bzw. direkt nutzen.

Im Fall der clientbasierten Systeme erfolgt die Kommunikation direkt zwischen den einzelnen Clients. Teilweise werden aber die Metadaten repliziert, wodurch es nicht mehr nötig ist, sämtliche Clients anzusprechen.

Weiterhin soll untersucht werden, in welcher Form die einzelnen Repositories ihre Informationen bereitstellen. Wichtig für die Föderation von Kursinhalten ist, dass möglichst kleine Kurseinheiten bereitgestellt werden. Siehe hierfür auch [MB03].

Im Fazit dieses Kapitels soll dargelegt werden, warum keines dieser Systeme ausreichend für das WIESELFRAMEWORK und somit eine eigene Implementation nötig ist.

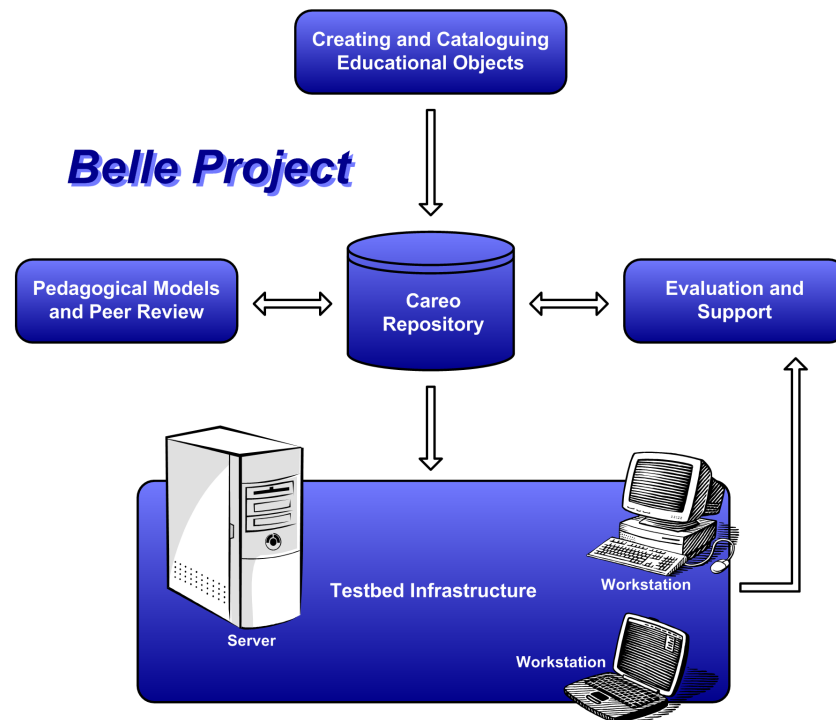


Abbildung 9: Architektur von BELLE

3.2. BELLE

Das BELLE-Projekt [Net01] wird finanziert über das CANARIE Learning Program [CAN93]. Die vier Aspekte die im Projekt betrachtet wurden, sind:

1. Erstellen und Katalogisieren von Lernobjekten,
2. Verwendung von pädagogische Modelle und Expertengutachten,
3. Evaluation und Support und
4. Bereitstellung einer Testbett-Infrastruktur.

Im ersten Punkt wurde zusammen mit anderen Lernprojekten das "Canadian Core" geschaffen, welches ein Standard zur Beschreibung von Inhalten ist, um effizient innerhalb der Repositories suchen zu können. Ein weiterer Anteil in diesem Punkt der Arbeit war die automatisierte Erzeugung von Metadaten und die Schaffung von Tools für die Erstellung von Metadaten.

Im zweiten Punkt ging es um die Begutachtung der Lerninhalte durch Experten und die Abschätzung der Qualität.

Für den dritten Punkt wurden Instrumente für die Evaluation erstellt und es wurde die Supportstruktur festgelegt.

Als Testbett wurden Client-Lernumgebungen, Server und “Content Repurposing Facilities” erstellt. Die Client-Lernumgebungen sind mobile Arbeitsrechner, die miteinander über broadcast verbunden sind. Die Server enthalten die Lerninhalte und können von dem Client angefragt werden. Die “Content Repurposing Facilities” dienen dem Digitalisieren und Markieren der Inhalte.

3.2.1. Peer Review

Um die Qualität der Kurse gewährleisten zu können, werden kanadische Ausbildungsinstitute bei der Entwicklung der Kurse einbezogen. In vielen Fällen, in denen versucht wird, die Qualität von Kursen zu verbessern, geht der erhöhte Zeit- und Kostenaufwand auf Kosten von anderen Aktivitäten. Die Bewertung der Qualität von Kursen erfolgt in den meisten Fällen durch Experten, die diese Beurteilung nur aufgrund ihrer Erfahrungen vornehmen. Es gibt keine ausreichende maschinelle Unterstützung für diesen Prozess.

Eine kritische Komponente des Evaluierungsprozesses ist das “Peer Review of Instructional Technology Innovation (PRITI)“-Projekt. Dieses Projekt beschäftigt sich mit zwei Aufgaben.

Die erste Aufgabe war die Untersuchung von Modellen und Ansätzen von Unterrichts-, akademischen Experten- und Lernobjektsbewertungen, sowie Untersuchungen im Bereich der wissenschaftlichen Lehre.

Als zweite Aufgabe wurden Modelle und Instrumente entwickelt, die den Partnerinstituten bei Expertenbewertungen helfen sollen. Als Unterstützung für die Entwicklung dieser Modelle und Instrumente wurden die Partnerinstitute zum begleitenden Testen der Entwicklungsstadien herangezogen.

3.2.2. Content Repurposing

In den meisten Fällen werden Kurse für einen bestimmten Zweck und eine bestimmte Umgebung geschaffen. Teile dieser Kurse können nur schwer für andere Kurse genutzt werden. Durch “Content Repurposing” sollen Lernobjekte so gestaltet werden, dass ein Wiederverwenden dieser Objekte erleichtert wird. Durch diese Bestrebung Lernobjekte mehrfach zu verwenden, kann das entsprechende Material kosteneffektiver, nachhaltiger und von einer höheren Qualität erstellt werden.

Für diese Aufgabe werden von den Partner des BELLE-Projektes Theorien, Modelle und Entwicklungswerkzeuge entwickelt. Ein Ziel ist es, einen Standard für die Entwicklung von wiederverwendbaren Lernmaterialien zu beschreiben. Um diese Bestrebung durchzusetzen, wird mit anderen Projekten aus dem Umfeld des eLearnings zusammengearbeitet.

3.3. Merlot

Das durch das MERLOT-Projekt [mer] entstandene LOR ist relativ groß, da es schon seit 1997 besteht. Die Suche nach Inhalten erfolgt über eine Webseite. Die Ergebnisse werden als Webseite zurückgegeben. MERLOT nutzt neben dem eigenen Repository auch die Repositories von Edna Online und SMETE. Ein wichtiger Punkt bei dem Projekt ist das "Peer Reviewing".

3.3.1. Peer Reviewing

Durch das "Peer Reviewing" soll es Instituten erleichtert werden, zu entscheiden, ob das Lernmaterial, welches sie gefunden haben, in ihre Kurse passt. Diese "Peer Reviews" werden nicht notwendigerweise von Autoren, sondern in den meisten Fällen von Nutzern durchgeführt.

Der Begutachtungsprozess von MERLOT orientiert sich an dem Modell für die Beurteilung von Stipendien. Jedes Lernmaterial eines bestimmten Wissensgebietes ist einem redaktionellen Gremium zugeordnet. Dieses Gremium entscheidet im ersten Schritt, welche Priorität das Lernobjekt für die Durchführung einer Beurteilung hat. Durch diese Priorisierung soll sichergestellt werden, dass die Lernobjekte mit der höchsten Qualität als Erstes beurteilt werden. Die Lernobjekte, die einen hohen Prioritätswert haben, werden als "accepted for review" gekennzeichnet. Sobald ein Experte das Kursmaterial sichtet, wird der Status des Objektes auf "under review" gesetzt.

Um nicht einseitige Beurteilungen zu erhalten, müssen mindestens zwei Beurteilungen von verschiedenen Fakultäten durchgeführt werden. Aus der Gesamtheit der Beurteilungen wird ein "composite review" erstellt, welcher auf der MERLOT-Webseite angezeigt wird.

Zurzeit gibt es vierzehn Gremien, die Beurteilungen von Kursmaterialien durchführen. Eine Liste dieser Gremien ist auf der MERLOT-Webseite zu finden.

Beim “Peer Review“ werden drei Bereiche betrachtet:

- Qualität des Inhalts,
- Potentielle Effektivität als Lehrinstrument und
- Benutzerfreundlichkeit.

Alle diese Bereiche werden getrennt voneinander betrachtet. Dabei erhält jeder Bereich eine Bewertung zwischen einem und fünf Sternen. Fünf Sterne entsprechen hierbei dem höchsten Wert. Damit ein Lernmaterial im LOR von MERLOT aufgenommen wird, muss es mindestens eine Gesamtbewertung von drei Sternen haben.

3.4. EdnA Online

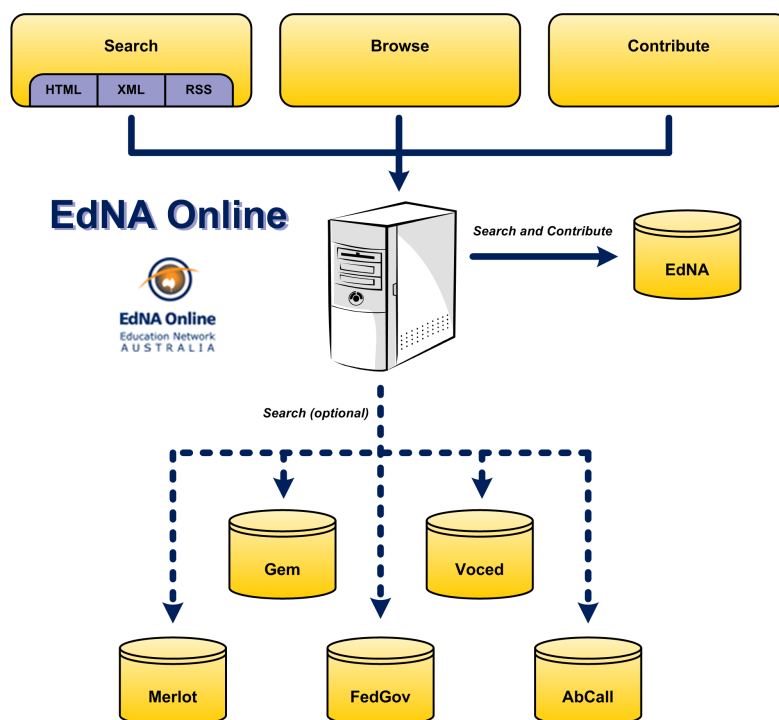


Abbildung 10: Architektur von Edna Online

Edna Online [edu03] ist ein Projekt des Bildungsministeriums Australiens. Es ermöglicht die Suche von Lerninhalten im Internet. Es gibt innerhalb dieses Projektes verschiedene Möglichkeiten der Suche und des Durchblätterns, des LORs.

Es gibt Anfragen über:

1. HTML,
2. RSS und
3. XML.

Die HTML-Anfrage wird auf der Webseite in Form einer Suchmaske angeboten. Die XML-Anfrage kann für die maschinelle Anfrage genutzt werden. Das dritte Format, RSS, ist ein Sonderformat für einen News Reader.

Die Anfrageparameter sind in allen drei Fällen die Gleichen. Für diese Arbeit ist nur die XML-Anfrage relevant, daher werden die nachfolgenden Erklärungen zu den einzelnen Parametern auf XML bezogen. Neben der Suche und dem Durchblättern werden noch Dienste für das Anzeigen von Überschriften und das Anzeigen der letzten hinzugefügten Ressourcen angeboten.

3.4.1. Parameter für die Anfrage

Parameter-name	Beschreibung	Wert(e)	Opt.?
q	Fachbegriffe getrennt durch +	Text	Nein
cs	casesensitiv	true, false(default)	Ja
items	Anzahl der Einträge pro Seite	1-200 (10 default)	Ja
kc	Suchbegriffverwendung	all, any, phrase	Ja
maxdesc	maximale Beschreibungslänge	1-3000 (1000 default)	Ja
mode	Darstellmodus für Javascript und HTML-Format URL (Titel und Beschreibung) Normal (Titel, URL und Beschreibung) Vollständig (Titel, Relevanz, Kategorie und Beschreibung)	url, normal, full(default)	Ja
mr	maximale Anzahl von Ergebnissen	50-200 (100 default)	Ja
popup	Öffnen der Links in einem neuen Fenster	true(default), false	Ja
showdesc	Anzeige der Beschreibungen	true (default), false	Ja
showsource	Anzeige der Quelle des Ergebnisses	false, true (default)	Ja
sr	Ressourcenquelle. Mehrere Werte erlaubt.	edna (default), gem, voted, merlot, fedgov, abcall	Ja
ss	Suchstrategie. Warten nur auf das erste Repository oder warten auf alle Repositories	waitfast (default), waitall	Ja
start	Startnummer der Ergebnisse auf dieser Seite	1-201 (1 default)	Ja

Der wichtigste Parameter ist der q-Parameter, welcher die zu suchenden Schlüsselwörter enthält. Durch die Parameter cs und kc kann die Suche weiter eingeschränkt werden. Weitere Einschränkungen können mittels des sr- und des ss-Parameters erfolgen. Während der sr-Parameter die verwendeten Repositories einschränkt bzw. erweitert, kann mit dem ss-Parameter, bei der Angabe mehrerer Repositories, festgelegt werden, wie das Warten auf die Ergebnisse abläuft. Es kann zwischen dem Warten auf alle Ergebnisse von allen Repositories und dem Warten auf das erste antwortende Repository gewählt werden.

3.4.2. Ergebnis der Anfrage

Das Ergebnis enthält in der XML-Version folgende Elemente.

summary

Dieses Element enthält Informationen über das Copyright, die Quelle der Inhalte, den Suchmodus, die Anzahl der gefundenen Ressourcen, die benötigte Zeit für die Anfrage und die Parameter der Anfrage.

sources

Hier sind weiterführende Informationen über die verwendeten Repositories aufgeführt. Für jedes Repository gibt es ein **source**-Element, welches den Titel, die URL und einen Bildlink enthält.

item

Dieses Element kann mehrfach auftreten und enthält die Informationen über die gefundenen Ergebnisse. Jedes **item**-Element hat ein Attribut **source** in dem die Repository-ID, welche beim **source**-Element angegeben wurde, enthält. Innerhalb des **item**-Elements wird der Titel, der Link zum Kurs, die Relevanz, eine Beschreibung und weitere Informationen angegeben. Zu erwähnen ist noch das Element **categories**, in welchem die Kategorien angegeben sind und unter welchem der Kurs im jeweiligen Repository zu finden ist.

3.5. SMETE

Die SMETE Open Federation [SME99] bietet Lerninhalte für die Bereiche:

1. Naturwissenschaft,
2. Mathematik,
3. Ingenieurwesen und
4. Technik.

Die Inhalte des LORs können über die Webseite durchsucht und durchgeblättert werden. Zu jeder Ressource kann das zugehörige LOM abgefragt werden. Die Metadaten werden nach dem IEEE LOM-Standard beschrieben. Neben der Anfrage direkt über die Webseite bietet das System eine Web Service-Schnittstelle, um auch maschinelle Anfragen zu ermöglichen.

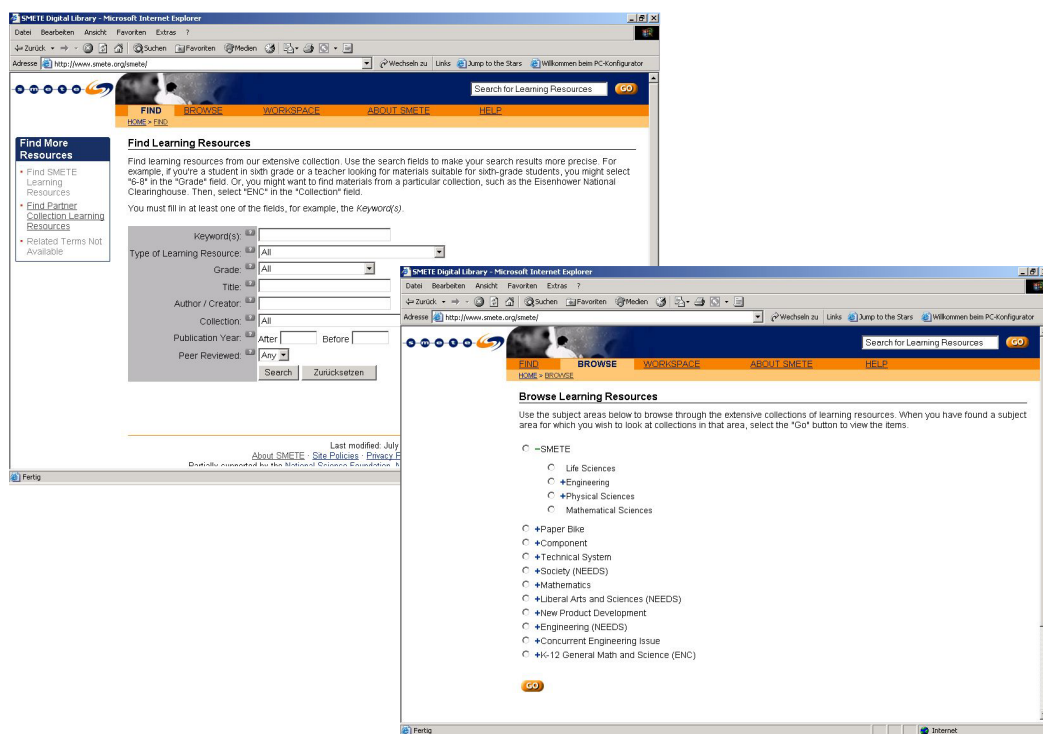


Abbildung 11: Suchen und Durchblättern in SMETE

3.5.1. Web Service-Schnittstelle

Die Web Service-Beschreibung (WSDL)

Die WSDL ist unter der Adresse:

<http://soap.smete.org/axis/services/urn:SmeteSearch?wsdl>

zu finden.

Die Anfrageparameter

1. key
2. q
3. start
4. maxresult
5. language

Der Key-Parameter wird momentan noch nicht verwendet. Er ist eingeführt, um in späteren Versionen das DRM zu unterstützen. Der Q-Parameter nimmt die eigentliche Anfrage auf. Es gibt für diesen Parameter zwei Arten der Darstellung.

Die unterstützten Elemente des IEEE LOM-Standards sind:

DC-Ed	IMS LOM	HTTP
DC.Title	General.Title	Title
DC.Subject	General.Keywords	Subject
	Classification.Keywords	Keywords
DC.Contributor	Lifecycle.Contribute, wenn Lifecycle.Contribute.Role=Any	Contributor
DC.Date	Lifecycle.Contribute.Date, wenn Lifecycle.Contribute.Role=Publisher	Date
DC-Ed.Audience	Educational.IntendedEndUserRole	Audience
N/A	Annotation.Description	Annotation

Tabelle 2: Unterstützte Elemente des LOM-Standards

Der erste Art der Darstellung der Parameter entspricht der Lucene-Anfragesyntax.

Beispiel:

```
general.title: "cell biology" AND educational.learningresourcetype: exercise
```

Der zweite Art entspricht dem LOM-Standard des IEEE. Die einzelnen Elemente werden um ein Attribut `boolean` erweitert, welcher die Werte `AND`, `OR` and `NOT` annehmen kann.

Beispiel:

```
<lom
  xmlns:ims_lom="http://www.imsglobal.org/xsd/imsmd_v1p2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://www.imsglobal.org/xsd/imsmd_v1p2
    http://www.imsglobal.org/xsd/imsmd_v1p2p2.xsd">
  <general>
    <title boolean="AND">
      <langstring>cell biology</langstring>
    </title>
  </general>
  <educational>
    <learningresourcetype boolean="AND">
      <langstring>exercise</langstring>
    </learningresourcetype>
  </educational>
</lom>
```

Der Start-Parameter teilt dem System mit, von welchem Startwert das Ergebnis geliefert werden soll. Der Maxresult-Parameter gibt die maximale Anzahl der Ergebnisse an. Mit dem Language-Parameter wird die Sprache der Ergebnisse festgelegt. Im momentanen Stand des Systems wird nur Englisch unterstützt, da nur englische Learning Objects im System verwaltet werden. Der Parameter folgt der XML Language Specification des W3C [Wor04b].

Das Ergebnis

Das Ergebnis kann auf zwei Arten formatiert sein. Das Schema der Standardformatierung ist im Anhang angegeben. Es entspricht in weiten Teilen den `ListRecords`-Element des OAI PMH[LSNW02].

Neben dem erfolgreichen Abschluss der Anfrage kann es zu Fehlern kommen, die als **Error**-Elemente zurückgegeben werden. Neben dem Element **badQuery** für eine falschformulierte Anfrage und dem Element **noRecordsMatch** für eine leere Ergebnismenge gibt es ein Element, welches wieder für zukünftige Entwicklungen im Bereich DRM vorhanden ist, mit Namen **notAllowed** für nicht erlaubte Zugriffe.

Die zweite Formatierung ist einfacher strukturiert, um das Arbeiten mit XSL zu erleichtern. Es besteht nur aus einer Liste von **record**-Blöcken.

3.6. Edutella

Das Edutella-Projekt [NDS02] stellt ein Framework zur Verfügung, um Metadaten innerhalb eines P2P-Netzwerkes zu nutzen. Diese Metadaten werden zur Suche der benötigten Daten verwendet.

Ziel ist es eine RDF-basierte Metadateninfrastruktur zu schaffen, die auf der Basis des JXTA-Frameworks [jxt99] arbeitet.

Ein wichtiger Service ist der Anfrageservice. Für diesen Service existiert das Edutella Common Data Model (ECDM), welches die Basis für die RDF-QEL-i bildet.

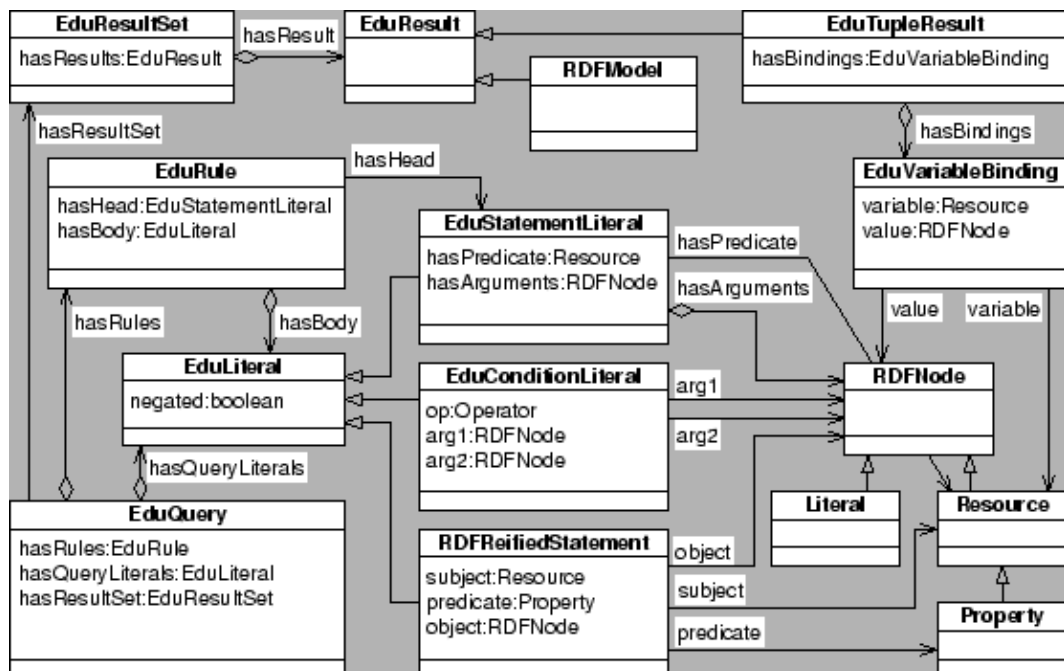


Abbildung 12: Edutella Common Data and Query Exchange Model (ECDM)

Folgende Dienste werden durch Edutella bereitgestellt:

1. Query Service,
2. Replication Service,
3. Mapping Service,
4. Mediation Service und
5. Annotation Service.

Der Query Service bietet die Möglichkeiten, standardisierte Anfragen und Suchen auf den Metadaten durchzuführen.

Der Replication Service stellt die Datenpersistenz und -verfügbarkeit sicher, sorgt für einen Auslastungsausgleich bei der Datenintegration und für die Konsistenz der Daten.

Der Mapping Service übersetzt das Metadatenvokabular der einzelnen Clients, um die Kompatibilität zwischen einzelnen Clients zu sichern.

Der Mediation Service definiert Sichten, welche die Daten der unterschiedlichen Metadatenquellen verbindet und auftretende Konflikte und Überlappungen auflöst.

Der Annotation Service kommentiert Material, das irgendwo in Edutella-Netzwerk gespeichert ist.

Im Weiteren soll näher auf den Query Service und auf den Mapping Service eingegangen werden, da diese beiden Teile auch beim Wiesel-Framework umgesetzt werden sollen.

3.6.1. Query Service

Der Query Service der Edutella-Infrastruktur bietet die Möglichkeit, innerhalb des Netzwerkes nach bestimmten Daten zu suchen.

Um möglichst viele Systeme ansprechen zu können, wurden in diesem Ansatz fünf verschiedene Anfragesprachstufen entwickelt. Die ersten drei sind zum Zeitpunkt dieser Arbeit bereits umgesetzt worden. Die anderen beiden liegen als Spezifikation vor.

Als Kriterien für die Gestaltung dieser Anfragestufen wurden folgende aufgestellt:

- Standardsemantik,
- hohe Aussagekraft,
- hohe Anpassbarkeit und
- Umwandlungsfähigkeit.

Die Standardsemantik ist besonders notwendig, um das vierte Kriterium umzusetzen. Die Beschreibungsemantik soll der RDF Semantik entsprechen.

Die hohe Aussagekraft soll ermöglichen, dass sowohl Clients mit einfacher Semantik direkt angesprochen werden können, als auch komplexere Systeme.

Die Anpassungsfähigkeit dient der Nutzung verschiedener Formalismen, wodurch es möglich wird sowohl Systeme, die auf relationalen Datenbanken basieren, als auch Systeme mit RDFS-Repositories und objektorientierten Datenbanken zu verwenden.

Die Umwandlungsfähigkeit ermöglicht die Verwendung von Wrappern, um einzelne Anfragesprachen ineinander zu überführen.

Die erste Stufe, RDF-QEL-1, ist die primitivste. An diese Anfragesprache kann jeder beliebige RDF-Graph übergeben werden. Die Antwort besteht aus dem gleichen Graphen mit zusätzlichen Informationen. Die Graphstruktur wird als logische, konjunktive Formel interpretiert, die auf einer Wissensbasis basiert. Disjunktiven Verknüpfungen sind in dieser Stufe nicht möglich, daher müssen diese durch mehrere Unteranfragen aufgelöst werden.

Die RDF-QEL-2 ist eine Erweiterung um Disjunktion gegenüber RDF-QEL-1. In dieser Stufe wird nicht mehr der RDF-Graph als Ganzes betrachtet, sondern die Elemente des RDF-Tripel werden in eine neue Struktur verpackt. Diese Struktur sind die **QueryStatements**. Neben den einzelnen Statements können auch komplexe, die durch **AND** und **OR** miteinander verknüpft sind, bearbeitet werden. Hierdurch wird die Konjunktion und Disjunktion verfügbar gemacht. Diese Anfrage kann leicht visualisiert werden. Für die Visualisierung der Anfragen stellt das Edutella-Netzwerk das Anfrageinterface "Conzilla" bereit. Die Darstellung orientiert sich an UML.

In der dritten Stufe der Anfragesprache existiert neben der Konjunktion und Disjunktion auch die Möglichkeit der Negation von Literalen.

In der Spezifikation der vierten Stufe sind neben den bisher implementierten Sachverhalten, auch die Rekursion zur Darstellung von Transitivität und die

linear-rekursive Anfragen enthalten. Durch diese zusätzlichen Eigenschaften soll diese Stufe konform zum SQL99-Standard gemacht werden.

In der letzten Stufe soll es zusätzlich möglich sein, beliebige rekursive Anfragen zu stellen und es soll ein einheitliches Minimalmodell erstellt werden, um eindeutige Ergebnisse zu bekommen.

Zusätzliche Erweiterungen der Anfragesprache sind die RDF-QEL-i-A - Stufen, mit denen Aggregationsfunktionen aus SQL2 ausgeführt werden können.

3.6.2. Mapping Service

Das Mapping verschiedener Anfragen erfolgt über spezielle Wrapper. Diese Wrapper gibt es bereits für die folgenden Anfragesprachen:

1. RQL,
2. TRIPLE,
3. SQL,
4. XPath/dbXML und
5. AmosQL.

Neben den Wrappern ist es auch nötig, so genannte Mediatoren einzusetzen. Die Mediatoren übernehmen die Rolle der Transformation der Ergebnisse, um eine kohärente Sicht auf die Ergebnisse zu ermöglichen. Diese Transformation erfolgt im Edutella-Netzwerk über einen zweistufigen Ansatz.

Einfache Mediatoren haben als Einschränkung, dass die komplette Anfrage von einem Client beantwortet werden muss. Komplexe Mediatoren können verteilte Anfragen an mehrere Repositories durchführen.

Für die Registration der Clients werden Informationen über das benutzte Schema mit eventuell auftretenden Wertebeschränkungen benötigt. Das Registrierungsformat entspricht dem RDF-QEL-1-Anfragelevel. Ein weiterer Wert, der bei der Registrierung angegeben werden muss, ist die unterstützte Anfragesprachstufe.

Der Netzwerkknoten empfängt Anfragen und sendet, aufgrund der Registrierungsdaten, die Anfragen an die Clients, mischt die Ergebnisse und sendet sie als eine Ergebnismenge zurück.

Die zweite Stufe kann verteilte und heterogene Anfragen verbinden. Weiterhin senden die Mediatoren Teilanfragen an verschiedene Repositories, vereinen die

Teilergebnisse und senden diese als einheitliche Ergebnismenge wieder an den Clients zurück.

3.7. Vergleich der Systeme und Fazit

	WIESELFramework	MERLOT	EdnA	Smete	Edutella
Bereitstellung der Daten über	Server	Server	Server	Server	P2P-Netz
Werden Kurse angeboten?	•	•	•	•	•
Werden SCO's angeboten?	•	-	-	-	•
Steht eine API für die Suche zur Verfügung?	•	-	•	•	-
Kann der Zugriff über Web Services erfolgen?	•	-	-	•	-(•)
Existiert eine Föderation von Daten?	•	•	-	•	-

Tabelle 3: Vergleich der untersuchten Systeme

Eine mögliche Nutzung von Webservices im Edutella-Netzwerk wird in [QN04] dargestellt.

Mittlerweile gibt es viele Ansätze für die Suche von Lerninhalten im Internet. In den meisten Fällen beschränkt sich die Suchfunktionalität auf Suchmasken im Browser. Diese Ansätze sind für die maschinelle Suche, wie sie für die Anforderungen des WIESELFrameworks erforderlich sind, nur eingeschränkt nutzbar. Während die Formulierung der Anfrage über HTTP noch relativ einfach umsetzbar ist, ist die Auswertung der Ergebnisse mit größeren Problemen behaftet.

Neben der Webanfrage bietet das SMETE-LOR auch eine Web Service-Schnittstelle, wodurch es wesentlich besser für die Nutzung zur Föderation geeignet ist. Leider ist das Ergebnis, wie auch in den meisten anderen untersuchten LOR's, auf einen Link zu den Lerninhalten beschränkt. In einigen Fällen verweisen diese Links nicht einmal direkt auf die gefundenen Kurse, sondern auf die Herstellerseiten der Kurse. Hierdurch ist die automatische Föderation auch mit diesem System nicht durchführbar.

Ein weiterer Ansatz ist die Nutzung von P2P-Netzen, um Kursinhalte auszutauschen. Das System, welches sich mit diesem Ansatz auseinandersetzt, ist das Edutella-Netzwerk. Es nutzt vorhandene P2P-Strukturen und erweitert diese, um die Fülle an Metadaten im eLearning-Umfeld nutzen zu können.

Ein weiterer Punkt der gegen die Nutzung von vorhandenen Systemen spricht, ist die zu geringe Modularisierung der Kurse. In den meisten Fällen können nur gesamte Kurse föderiert werden. Durch diese Abgeschlossenheit der Kurse muss

nicht auf die didaktische Anordnung innerhalb der Teilkurse geachtet werden. Es ist aber nicht möglich relevante Daten von irrelevanten zu trennen.

Die Föderation zwischen verschiedenen Systemen erfolgt nur mittels der Anfrage an die Webschnittstelle und der Auflistung der Ergebnisse innerhalb des Systems. Die Darstellung der Ergebnisse wird an das Layout des LMS angepasst.

4. Lösungsansatz

Nachdem im vorherigen Kapitel Systeme verglichen wurden, die sich teilweise mit der Föderation von eLearning-Kursen beschäftigen, sollen in diesem Kapitel die in der Arbeit entwickelten Lösungsvorschläge erläutert werden. Hier geht es um theoretische Konzepte, die die Anforderungen der Arbeit erfüllen können. Die gefundenen Konzepte bilden dann die Basis für die Umsetzung des Prototypen.

Wie in Kapitel 1 im Abschnitt 1.4 auf Seite 5 erläutert wurde, geht es in der Arbeit um die Nutzung von semantischen Diensten zur Erstellung von Kursen. Die Inhalte für diese Kurse stammen dabei nicht aus einem eLearning-System, sondern aus verschiedenen Learning Object Repositories und anderen Datenquellen, die Wissen in beliebiger Form gespeichert haben.

In den nachfolgenden Abschnitten wird darauf eingegangen, wie und welche semantischen Dienste genutzt werden können, um eine Föderation der verschiedenen Wissensarten zu ermöglichen. Weiterhin wird untersucht, welche Voraussetzungen gegeben sein müssen, um auf automatischen Wege verschiedene Wissensseinheiten miteinander zu einem Kurs zu verbinden. Dafür werden in einem weiteren Abschnitt didaktische Konzepte besprochen, welche für eine Föderation von Kursteilen zu einem Gesamtkurs beachtet werden müssen. Zusätzlich wird hier ein Ansatz aufgezeigt, wie didaktische Informationen über die Kursteile benutzt werden können, um die Kursteile innerhalb des Kurses didaktisch korrekt zu positionieren.

Nach der Beschreibung dieser grundlegenden Überlegungen wird ein Ansatz entwickelt, der es ermöglicht, aus verschiedenen eLearning-Systemen (insbesondere LOR's) Inhalte zu vereinen, so dass ein Kurs entsteht, welcher die Anforderungen des Nutzers möglichst gut erfüllt.

In einem weiteren Abschnitt wird dann ein Konzept vorgestellt, durch welches Inhalte aus anderen Wissensquellen in den Kurs aufgenommen werden können.

Die so entstandene Menge von Lerneinheiten wird auf ihre Kohärenz im Hinblick auf die Darstellung untersucht. In diesem Abschnitt werden Ideen entwickelt, wie diese Inkonsistenz der Darstellungform beseitigt werden kann.

Die so entstandenen Lerneinheiten werden in einem nächsten Schritt zu einem Kurs zusammengefasst und dem Lerner über das Lernmanagementsystem zur Verfügung gestellt.

Im Abschnitt über WIESELMOBILE werden Konzepte diskutiert, welche es ermöglichen, Kurse so anzupassen, dass sie in mobilen Umgebungen betrieben werden können. Weiterhin wird hier untersucht, wie Informationen über das Endgerät des Nutzers und die Umgebung des Nutzers an das Föderationssystem übergeben werden können.

Den Abschluss dieses Kapitels bildet eine Zusammenfassung der gefundenen Konzepte.

Im sich anschließenden Abschnitt soll aber erst die Architektur aufgezeigt werden, in welche sich die zu ermittelnden Konzepte einfügen.

4.1. Architektur

Die Architektur des WIESEL-Projekts, in die diese Arbeit eingebettet ist, dient der Bereitstellung von eLearning-Kursen über das Internet. Für dieses Projekt wurden verschiedene Module entwickelt, welche die Grundfunktionalität eines Lernmanagementsystems erweitern.

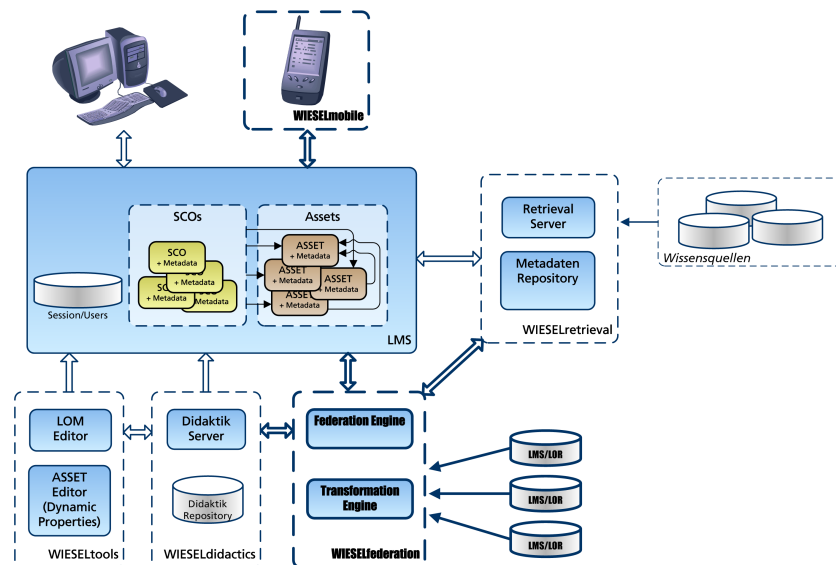


Abbildung 13: Die Architektur des WIESELFrameworks mit Erweiterungen aus der Arbeit

In der Abbildung 13 sind die Erweiterungen der Arbeit durch dickere Linien hervorgehoben. Zum Vergleich kann die Abbildung 1 im Kapitel 1 auf Seite 2 herangezogen werden. Wie in der Abbildung 13 dargestellt wird, beschäftigt

sich diese Arbeit mit den Modulen WIESELFEDERATION und WIESELMOBILE.

WIESELFEDERATION bietet die Möglichkeit verschiedene Kursteile aus verschiedenen LOR's und Wissensquellen zu einem Kurs zusammenzusetzen.

Mittels WIESELMOBILE wird eine Nutzung des WIESELFEDERATION durch mobile Endgeräte ermöglicht.

Das Modul WIESELFEDERATION wurde in kleinere Untermodule unterteilt, um eine Trennung der verschiedenen Aufgaben vornehmen zu können. Diese Unterteilung wird in Abbildung 14 dargestellt.

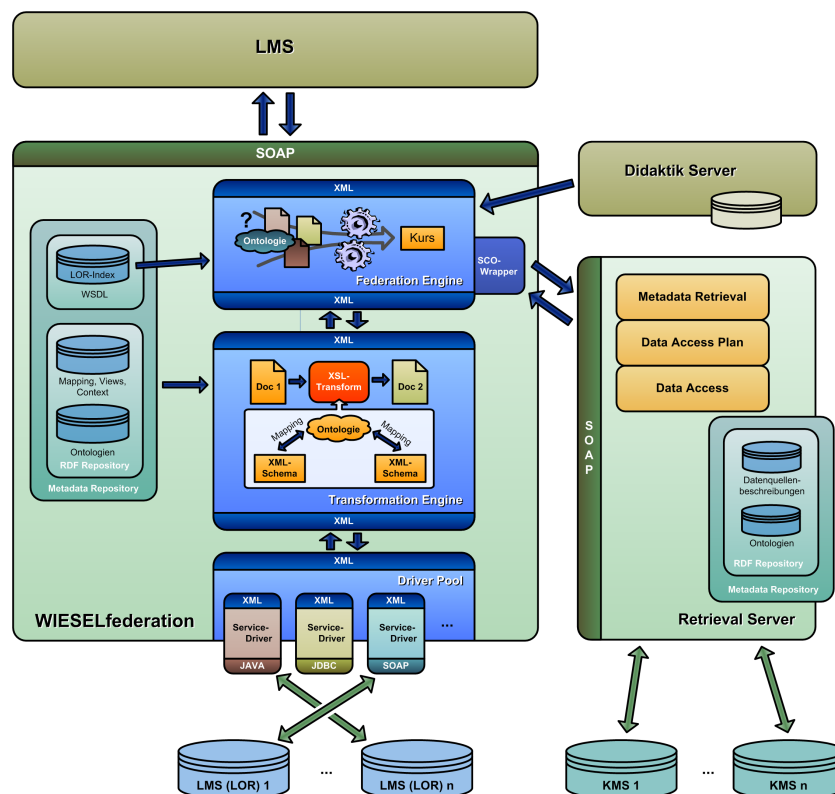


Abbildung 14: Architektur der WIESELFederation

Die Komponenten, welche innerhalb der Föderation eine Rolle spielen, sind

- die *Federation Engine*,
- die *Transformation Engine* und
- die *Course Enrichment Engine*.

Die *Federation Engine* generiert einen Kurs, welcher die Anforderungen des Lernenden erfüllt. Hierzu generiert diese Komponente Suchanfragen, um das benötigte Wissen durch Kursteile der verschiedenen LOR's bereitzustellen. Weiterhin werden die durch die Begriffe, welche durch die Kursteile erlernt werden, an die *Course Enrichment Engine* weitergegeben, um den Kurs durch Inhalte aus anderen Wissensquellen zu ergänzen. Das Ergebnis dieser Komponente ist ein Kurs, welcher im übergeordneten Lernmanagementsystem bereitgestellt wird. Die *Federation Engine* wird im Abschnitt 4.6 auf Seite 61 detailliert behandelt.

Da die Schnittstellen zur Anfrage an die verschiedenen LOR's heterogen sind, ist es nötig, eine Übersetzung der Anfrage der *Federation Engine* vorzunehmen. Die *Transformation Engine* übernimmt diese Aufgabe. Das von dieser Komponente erzeugte Ergebnis ist eine Übersetzung der Antwort des entsprechenden LORs, so dass die *Federation Engine* diese verarbeiten kann. Im Abschnitt 4.5 auf Seite 54 wird genauer auf den Prozess der Transformation eingegangen.

Die Komponente *Course Enrichment Engine* leitet die Anfragen der *Federation Engine* an die Wissensquellen weiter. Die Ergebnisse werden durch Metadaten angereichert und zu einer Kurseinheit verpackt, so dass das LMS in der Lage ist, diese Informationen anzuzeigen. Über diese Komponente sind weitere Informationen im Abschnitt 4.6.6 auf der Seite 67 zu finden.

Die Untersuchungen für das Modul WIESELMOBILE sind im Abschnitt 4.7 auf Seite 69 dargestellt.

Im nächsten Abschnitt sollen Überlegungen angestellt werden, wie semantische Dienste genutzt werden können, um eine Föderation von Kurseinheiten aus verschiedenen LOR's und Wissensquellen zu ermöglichen. Hierfür werden Untersuchungen über Ontologien und Metadaten, welche die Kurseinheiten beschreiben, dargelegt.

4.2. Ontologien und Metadaten

In diesem Abschnitt wird die Nutzung von Ontologien untersucht, um Probleme, welche durch Heterogenität entstehen, zu lösen. Die Ontologien dienen hierbei als eindeutige Wissensnetze, die in abgeschlossenen Fachdomänen die Beziehung zwischen einzelnen Begriffen herstellen.

Ein weiterer Punkt dieses Abschnittes stellt eine Spezifikation von Metadaten, welche für die Beschreibung von Lerneinheiten benutzt wird, vor. Zusätzlich

genutzt. Um die Ontologie sowohl für die Übersetzung der Anfragen der *Federation Engine* in die verschiedenen Anfrageformate der LOR's zu nutzen, wie auch für Übersetzungen der Antworten der LOR's in für die *Federation Engine* verständliche Antworten, wurde eine Oberklasse **LORCommunication** eingeführt. Die Unterklassen dieser Klasse sind die beiden Klassen **LORRequest** und **LORResponse**.

Die Klasse **LORResponse** wird für die Transformation der Antworten der verschiedenen LOR's in die der *Federation Engine* genutzt. Innerhalb dieser Klasse existiert eine Beziehung zu der Klasse **SCOURI**, welche die Lage der SCO im Internet präsentiert. Weiterhin existiert eine Klasse **Metadata**, welche die Metadaten der SCO repräsentiert und diese für Föderation bereitstellt.

Die Klasse **LORRequest** dient der Übersetzung der Anfrage der *Federation Engine* in die entsprechenden Anfragen an die unterschiedlichen LOR's. Die Unterklassen sind an den LOM-Standard angelehnt. Dadurch entstehen die neun Unterklassen, die dieser Spezifikation entsprechen. Diese sind:

- General
- Life Cycle
- Meta-Metadata
- Technical
- Educational
- Rights
- Relation
- Annotation und
- Classification.

Jede dieser Unterklassen hat Beziehungen zu anderen Klassen, welche die einzelnen Elemente der Metadaten repräsentieren.

Die Beschreibung der Ontologie erfolgt in der Beschreibungssprache OWL [Wor04c], um direkte Zugriffsmöglichkeiten auf die Beziehungen zwischen den Klassen zu haben. Durch OWL ist es möglich die komplette Klassenhierarchie darzustellen.


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema#">

  <owl:Class rdf:about="http://www.zgdv.de/ontologies/lom#LORCommunication">
    <rdfs:label>LORCommunication</rdfs:label>
  </owl:Class>
  <owl:Class rdf:about="http://www.zgdv.de/ontologies/lom#Description">
    <rdfs:label>Description</rdfs:label>
  </owl:Class>
  <owl:Class rdf:about="http://www.zgdv.de/ontologies/lom#MainKeyword">
    <rdfs:label>MainKeyword</rdfs:label>
    <rdfs:comment><![CDATA[general.keyword]]></rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="http://www.zgdv.de/ontologies/lom#Keyword">
    <rdfs:label>Keyword</rdfs:label>
  </owl:Class>
  <owl:Class rdf:about="http://www.zgdv.de/ontologies/lom#LORResponse">
    <rdfs:label>LORResponse</rdfs:label>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://www.zgdv.de/ontologies/lom#LORCommunication"/>
    </rdfs:subClassOf>
  </owl:Class>

</rdf:RDF>
```

Abbildung 16: Klassendefinition in OWL

Für die Beschreibung von Klassen in OWL wird das Element `owl:Class` benutzt. Dieses Element ist eine Unterklasse von `rdfs:Class`. Um die Klasse eindeutig zu identifizieren, gibt es das Attribut `rdfs:about`. Mittels des Elements `rdfs:label` wird eine Beschreibung der Klasse gegeben. Das Element `rdfs:subClassOf` wird genutzt, um eine Unterklassenbeziehung anzugeben. Ein Beispiel für die Beschreibung von Klassen mittels OWL ist in Abbildung 16 zu finden. Es zeigt verschiedene Klassendefinitionen der Ontologie. Im letzten Beispielement wird die Unterklassenbeziehung von `LORResponse` zur Klasse `LORCommunication` dargestellt.

Neben dem Unterklassen-Konzept gibt es in OWL mehrere Möglichkeiten zur Definition von Beziehungen zwischen zwei Objekten bzw. zwischen Klassen und XML-Datentypen.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema#">

  <owl:ObjectProperty rdf:about="http://www.zgdv.de/ontologies/lom#hasTaxonPath">
    <rdfs:label>hasTaxonPath</rdfs:label>
    <rdfs:domain>
      <owl:Class rdf:about="http://www.zgdv.de/ontologies/lom#LearningObject">
        </owl:Class>
    </rdfs:domain>
    <rdfs:range>
      <owl:Class rdf:about="http://www.zgdv.de/ontologies/lom#TaxonPath">
        </owl:Class>
    </rdfs:range>
  </owl:ObjectProperty>

</rdf:RDF>
```

Abbildung 17: Definition von Objekteigenschaften in OWL

Die Beziehung zwischen verschiedenen Objekten erfolgt mittels des Sprachelements `owl:ObjectProperty`. Die Klasse, für die die Eigenschaft existiert, wird durch das Element `rdfs:domain` definiert. Der Wertebereich dieser Eigenschaft wird mit dem Element `rdfs:range` bestimmt. In Abbildung 17 wird diese Beschreibung beispielhaft aufgezeigt. Es wird anhand dieses Beispiels beschrieben, dass die Eigenschaft `hasTaxonPath` für jedes Objekt der Klasse `LearningObject` existiert. Diese Eigenschaft ist ein Objekt der Klasse `TaxonPath`.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema#">

  <owl:DatatypeProperty rdf:about="http://www.zgdv.de/ontologies/lom#responseRankingValue">
    <rdfs:label>responseRankingValue</rdfs:label>
    <rdfs:domain>
      <owl:Class rdf:about="http://www.zgdv.de/ontologies/lom#LORResponse">
        </owl:Class>
    </rdfs:domain>
    <rdfs:range>
      <xsd:integer/>
    </rdfs:range>
  </owl:DatatypeProperty>

</rdf:RDF>
```

Abbildung 18: Definition von Datentypeneigenschaften in OWL

Um die Beziehung zwischen Klassen und XML-Datentypen zu beschreiben, wird das Element `DatatypeProperty` angewendet. In dem Element `rdfs:domain` wird wieder die zu dieser Beziehung gehörende Klasse erfasst. Das Element `rdfs:range` enthält den XML-Datentyp. Das zugehörige Beispiel ist in Abbildung 18 zu sehen. Im Beispiel wird beschrieben, dass jedes `LearningResponse`-Objekt eine Eigenschaft `responseRankingValue` hat, welches vom Typ `xsd:integer` ist.

Diese aufgezeigten Beschreibungselemente sind keine vollständige Beschreibung der Ontologiesprache. Sie zeigen nur die in der Arbeit benutzten Elemente.

Weitere Konzepte sind Untereigenschaften und Restriktionen.

Die Nutzung der Ontologie wird im Abschnitt *Transformation Engine* auf Seite 54 erläutert.

Fachontologie

Die Fachontologie enthält eine Beschreibung der Begriffe. Weiterhin werden die Beziehungen der Begriffe untereinander dargestellt. Die in der Arbeit verwendeten Ontologien entstammen dem Fachgebiet Biologie, da das Anwendungsszenario für die Arbeit die Kurserstellung für Studenten der Zellbiologie ist.

Zu den Beziehungen, die für die Arbeit genutzt wurden, zählen die Ober- und Unterklassenbeziehungen und die Synonymbeziehungen.

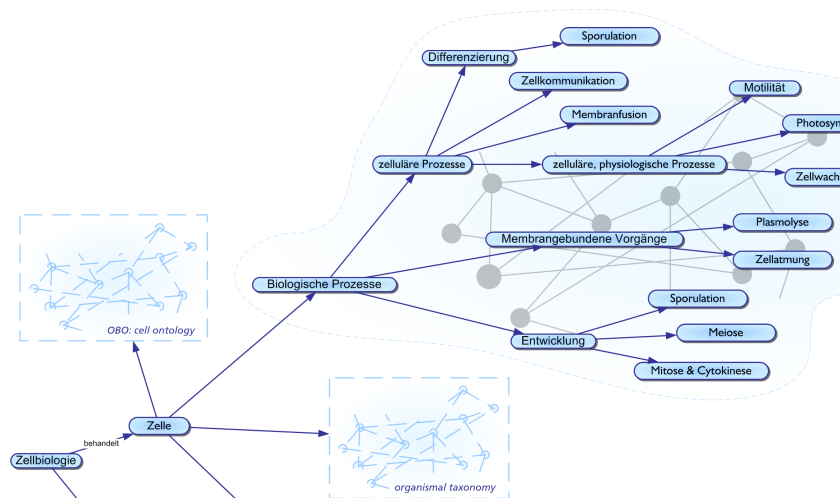


Abbildung 19: Ausschnitt aus der Zellbiologie-Ontologie

Ein abstrahierter Ausschnitt aus den verwendeten Ontologien ist in Abbildung 19 zu finden. Neben den reinen Begriffen aus dem Gebiet Zellbiologie wird in der Abbildung auch die Möglichkeit der Verknüpfung von Ontologien dargestellt.

Weiterführende Informationen können im Abschnitt über die *Federation Engine* auf Seite 61 nachgelesen werden.

4.2.2. Metadaten der Lernobjekte

Die Beschreibung der Lernobjekte durch Metadaten ermöglicht die Suche nach bestimmten Inhalten durch Maschinen. Es wird damit möglich, neben der rein inhaltlichen Suche, auch Lernobjekte zu finden, die z. B. in einem bestimmten Format vorliegen oder von einem bestimmten Autor erstellt wurden.

Metadaten sind somit Grundvoraussetzung für eine automatische Föderation von Lernobjekten zu einem gesamten Kurs.

Um diese Anfragen möglich zu machen, ist es nötig, dass alle in den LOR's gespeicherten SCO's mittels Metadaten beschrieben sind. Die beiden für den Prototypen dieser Arbeit verwendeten LOR's nutzen für die Bereitstellung der Metadaten den LOM-Standard. Dieser Standard ist momentan sehr weit verbreitet. Er wird bevorzugt von Kursen genutzt, die sich an die SCORM-Spezifikation halten. SCORM ist ein wichtiger Bezugspunkt innerhalb der eLearning-Welt geworden. Weiterhin enthält dieser Standard viele Elemente, die in der Arbeit genutzt werden. Benötigte Elemente können der Spezifikation leicht hinzugefügt werden, da die Metadaten in XML vorliegen. Durch die Beschreibung in XML ist es weiterhin möglich, sehr einfach auf die entsprechenden Daten zuzugreifen.

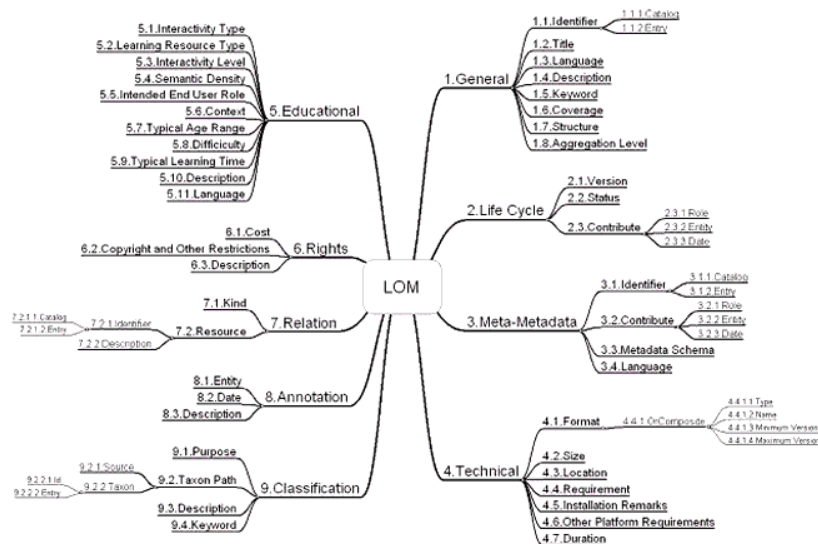


Abbildung 20: LOM-Standard des IEEE[IEE02]

In Abbildung 20 sind alle Elemente des LOM-Standards graphisch dargestellt.

Bei anderen Metadaten-Spezifikationen, wie dem CanCore[FMH] und dem Dublin Core[dub], müssen auch die zusätzlich benötigten Daten integriert werden, um sie für die Föderation nutzbar zu machen.

Soll eine andere Metadaten-Spezifikation aus dem Bereich des eLearnings eingesetzt werden, müssen die Transformationen an die jeweilige Spezifikation angepasst werden. Weitere Informationen sind im Abschnitt über die *Transformation Engine* ab Seite 54 zu finden.

Um eine Nutzung der Ontologien zu ermöglichen, müssen die Metadaten in denselben Sprachen verfasst sein, wie die Klassenbezeichner der Ontologien. Der LOM-Standard unterstützt mehrsprachige Metadaten, womit es möglich ist, Ontologien zu nutzen, welche in verschiedenen Sprachen verfasst sind.

Wie bereits erwähnt, ist es nötig, Erweiterungen der Metadaten vorzunehmen, um eine Föderation möglich zu machen. Die verwendeten Erweiterungen sind der Tabelle 4 zu entnehmen.

Name	Beschreibung	Knoten in LOM
Vorwissen	benötigtes Wissen, um die SCO benutzen zu können	classification/previousknowledge
Ergebnisse	Wissen, welches durch die SCO vermittelt wird	classification/learningresults
Didaktisches Modell	verwendetes didaktisches Modell	educational/didacticmodel

Tabelle 4: Erweiterung der LOM-Spezifikation

Um eine Föderation von SCO's durchführen zu können, sind bestimmte Metadaten nötig. Diese sind:

Name	Beschreibung	LOM-Element
Hauptschlüsselwort(e)	Schlüsselworte des Fachgebiet	general/keyword
Ziele	Ziele, die durch die SCO erreicht werden	classification/learningresults
Vorwissen	Wissen, welches zur Durchführung der SCO nötig ist	classification/ previousknowledge

Tabelle 5: Notwendige Elemente für die Föderation

Wie in der Tabelle 5 zu erkennen ist, werden zwei der vorgenommenen Erweiterungen für die Föderation benötigt. Das **Hauptschlüsselwort** dient der Verringerung der Anzahl der Suchanfragen, in dem mittels dieses **Hauptschlüsselwortes** eine Vorauswahl der LOR's erfolgt. Die genaue Beschreibung der Verwendung des **Hauptschlüsselwortes** erfolgt im Abschnitt *Federation Engine* ab Seite 61.

Nachfolgend ein Beispiel für eine Metadatenbeschreibung:

```
<lom>
  <general>
    ...
    <keywords>
      <langstring xml:lang="en-US">cell biology</langstring>
    </keywords>
    ...
  </general>
  ...
  <classification>
    <keyword>
      <langstring xml:lang="de-DE">Zellzyklus</langstring>
      <langstring xml:lang="de-DE">Plasmodium</langstring>
      ...
    </keyword>
    <learningresult>
      <langstring xml:lang="en-GB">cell cycle</langstring>
      <langstring xml:lang="de-DE">Zellzyklus</langstring>
    </learningresult>
    <learningresult>
      <id>&G0;0007067</id>
      <langstring xml:lang="en-GB">mitosis</langstring>
      <langstring xml:lang="de-DE">Mitose</langstring>
    </learningresult>
    ...
    <prerequisite>
      <langstring xml:lang="en-GB">light microscopy</langstring>
      <langstring xml:lang="de-DE">Lichtmikroskopie</langstring>
    </prerequisite>
    <prerequisite>
      <id>&G0;0005576</id>
      <langstring xml:lang="en-GB">MRNA</langstring>
      <langstring xml:lang="de-DE">MRNA</langstring>
    </prerequisite>
    ...
  </classification>
  ...
</lom>
```

Weitere Informationen, die für die Suche nach bestimmten SCO's vorhanden sein sollten, sind:

Name	Beschreibung	LOM-Element
Titel	Titel der SCO	<code>general/title</code>
Schlüsselwort(e)	Schlüsselworte, die den Inhalt der SCO widerspiegeln	<code>classification/keyword</code>
Beschreibung	Beschreibung der SCO	<code>classification/description</code>
Sprache	Sprache der Inhalte der SCO	<code>general/language</code>
Didaktisches Konzept	Konzept, nach welchem die SCO didaktisch aufgebaut wurde	<code>educational/didacticmodel</code>
Lebenslauf	Autor und Version der SCO	<code>lifecycle</code>
Lerninformationen	Informationen über die Bedingungen zum Nutzen der SCO	<code>educational</code>
Technische Details	Informationen über die verwendete Technik innerhalb der SCO	<code>technical</code>

Tabelle 6: Wünschenswerte Elemente für die Föderation

Die drei letzten Punkte innerhalb der Tabelle 6 sind Gruppen von Elementen. Die aufgeführten Informationen sind nicht notwendig für die Suche aber können diese wesentlich vereinfachen, da die Menge der SCO's schneller eingeschränkt werden kann. Weiterhin kann mittels dieser Daten ein Ranking der Ergebnismengen vorgenommen werden. Laut einer Untersuchung von Norm Friesen[Fri04] sind folgende Metadaten-Elemente, die am meisten genutzt werden.

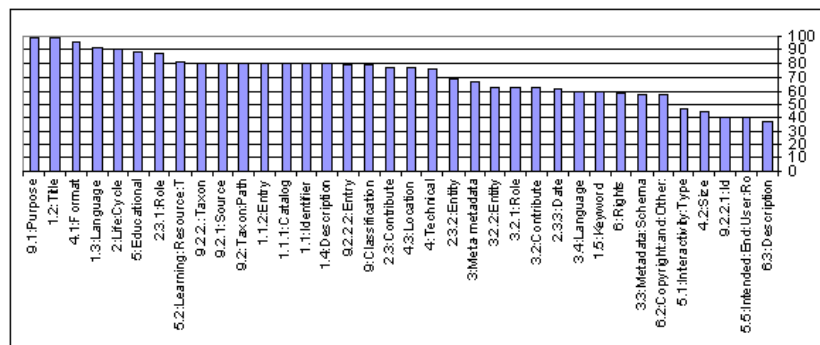


Abbildung 21: Hauptsächlich verwendete Metadaten[Fri04]

Beim Vergleich der Abbildung 21 mit den Tabellen 5 und 6 wird ersichtlich, dass viele Autoren die benötigten Elemente in ihren SCO's definieren.

Besonders im Bereich WIESELMOBILE sind die zusätzlichen Informationen über die technische Ausstattung der Geräte nötig. Informationen über die verwendeten Techniken und Formate sind in der Elementengruppe `technical` im LOM-Standard enthalten. Weiterführende Informationen, wie die technischen

Daten eines Gerätes gespeichert und verarbeitet werden, sind im Abschnitt WIESELMOBILE ab Seite 69 zu finden.

Kohärenz im Layout der verschiedenen SCO's wird produziert, um dem Nutzer ein einheitliches Aussehen des Kurses zu präsentieren und ihm dadurch das Lernen zu erleichtern. Dies erfolgt durch eine Transformationspipeline. Eine andere Pipeline wird genutzt, um im Aufgabenbereich WIESELMOBILE, die technischen Gegebenheiten der einzelnen Endgeräte zu berücksichtigen. Weitere Informationen über die Transformationspipeline können im Abschnitt WIESELMOBILE ab Seite 69 nachgelesen werden.

Ein weiterer Punkt, welcher bei der Föderation berücksichtigt werden muss, ist die Berücksichtigung von didaktischen Modellen. Um auf didaktische Vorlieben des Nutzers eingehen zu können, müssen bestimmte Informationen über das verwendete didaktische Modell erfasst werden. Dieser Punkt wird im nächsten Abschnitt vertieft.

4.3. Didaktische Konzepte

Als Betrachtungsgrundlage dienen die Makromodelle der Webdidaktik. Diese Makromodelle sind im Kapitel 2 im Abschnitt 2.1.5 auf Seite 11 beschrieben und graphisch in der Abbildung 3 dargestellt.

Die Makromodelle werden genutzt, um die SCO's, die durch die Suchanfragen erhalten wurden, zu einem Kurs zu föderieren. Zusätzlich werden die SCO's, die durch die *Course Enrichment Engine* erzeugt wurden, in den Kurs integriert.

Drei Modelle dieser Modellklasse sind:

- induktives Modell,
- deduktives Modell und
- Guided Tour.

In der Arbeit werden nur die beiden oberen Modelle verwendet, da das letzte Verfahren nur für fest definierte Kurse möglich ist. Das induktive Modell beginnt mit konkreten Sachverhalten und verallgemeinert diese innerhalb des Kurses. Das Gegenteil dazu ist das deduktive Modell. Hier wird der Kurs ausgehend vom Allgemeinen zum Konkreten aufgebaut. Die meisten Menschen nutzen zum Lernen das deduktive Modell.

In der Arbeit wird durch den Benutzer festgelegt, nach welchem didaktischen Makromodell er lernen möchte. Um eine möglichst gute Nutzung des jeweiligen

Modelles zu erreichen, wird dieses durch die Heuristik abgetestet. Dazu wird das zusätzlich eingefügte Metadaten-Element `didaktikmodel` verwendet.

Nachdem in den vorangegangenen Abschnitten allgemeine Konzepte untersucht wurden, welche für die Föderation von Lernobjekten aus heterogenen System unter der Zuhilfenahme von semantischen Diensten nötig sind, werden in den folgenden Abschnitten die notwendigen Ansätze für die Umsetzung der verschiedenen Module erörtert. Der nächste Abschnitt soll das Modul WIESELFEDERATION mit den einzelnen Komponenten darstellen, bevor in den darauf folgenden Abschnitten die einzelnen Komponenten detailliert beschrieben werden.

4.4. WIESELFederation

Das Modul WIESELFEDERATION dient der Föderation von Lernobjekten aus verschiedenen LOR's zu einem kompletten Kurs. Zusätzlich werden Erweiterungen des Kurses durch Hinzufügen von Daten aus anderen Wissensquellen durchgeführt. Das Zusammenfügen der Lernobjekte zu einem Kurs erfolgt nach didaktischen Gesichtspunkten, welche sich an den Vorgaben des Nutzers orientieren.

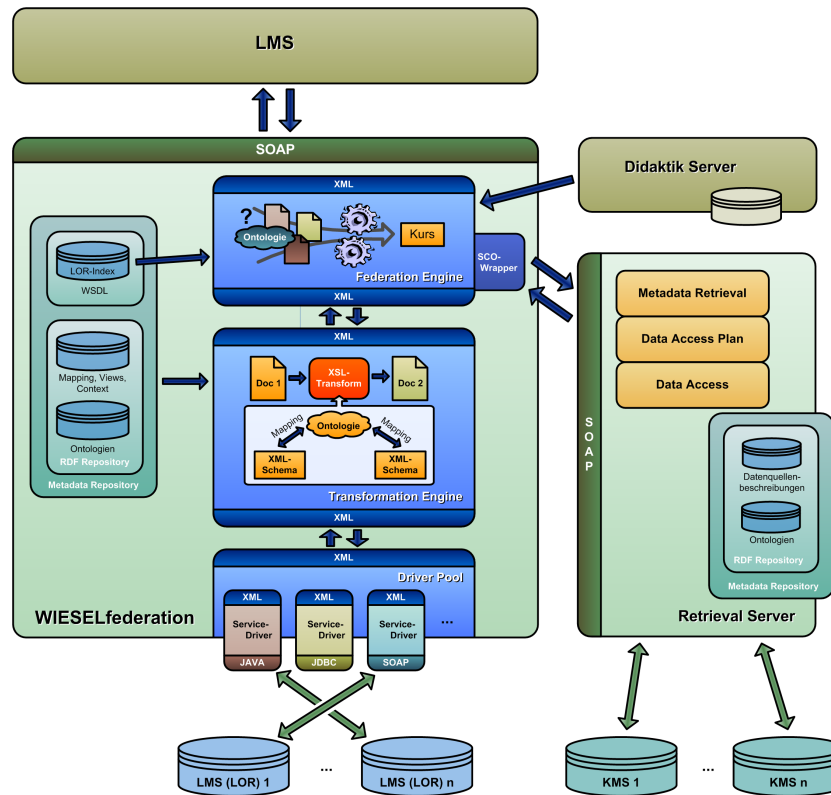


Abbildung 22: Darstellung der Komponenten von WIESELFEDERATION

In Abbildung 22 werden die drei Komponenten von WIESELFEDERATION dargestellt.

Die Hauptkomponente ist die *Federation Engine*. In der *Federation Engine* werden die Anfragen erzeugt, welche an die *Transformation Engine* übergeben werden, die sie dann an die LOR's übergibt. Eine weitere Aufgabe der *Federation Engine* ist die Weiterleitung der durch den Kurs erlernten Begriff an die *Course Enrichment Engine*, um zusätzliche Daten aus Wissensquellen zu erhalten. Die erhaltenen Kursteile müssen unter Zuhilfenahme von didaktischen Modellen zu einem Gesamtkurs zusammengesetzt werden.

Die *Transformation Engine* ist für die Weiterleitung der Anfragen an die verschiedenen LOR's zuständig. Um diese Aufgabe erfüllen zu können, muss sie in der Lage sein, die Heterogenitäten in den Schnittstellen zu beseitigen.

In der *Course Enrichment Engine* werden Anfragen an bestehende Wissensquellen durchgeführt, um den Kurs, welcher aus den Lernobjekten der verschiedenen LOR's besteht, mit zusätzlichem Wissen anzureichern.

4.5. Transformation Engine

Wie im Kapitel 1 im Abschnitt 1.3 auf Seite 4 beschrieben wurden, wird für die Umsetzung der *Transformation Engine* der Gatewayansatz aus der Diplomarbeit von Matthias Schulz [Sch03] verwendet. Die Aufgabe der *Transformation Engine* ist die Übersetzung der Anfragen der *Federation Engine* in die Anfragen der genutzten LOR's. Weiterhin wird die Anfrage über die Web Service-Schnittstelle der LOR's weitergereicht. Die Antwort der LOR's muss wieder zurücktransformiert werden, damit die *Federation Engine* die Daten nutzen kann.

In den Unterabschnitten werden die Funktion des Gateways erklärt und Untersuchungen vorgenommen, wie die Semantik der Anfragen berücksichtigt werden kann.

4.5.1. Gateway

Die Auflösung der Heterogenitäten der einzelnen Systeme, die durch das Gateway bedient werden, erfolgt nicht durch ein direktes Mapping, sondern durch die Nutzung einer Ontologie als Zwischenschritt. Die Ontologie ist allgemeingültig und vollständig, um alle Anfragekonzepte abbilden zu können.

Der Prozess der *Transformation Engine* läuft in drei Schritten ab. Diese sind:

1. die Übersetzung der Anfrage der *Federation Engine*,
2. das Senden der transformierten Anfragen an die LOR's und
3. die Übersetzung der Antworten der LOR's.

Die Ontologie wird sowohl im ersten Schritt für die Übersetzung der Anfragen der *Federation Engine* verwendet, als auch für die Übersetzung der Antworten der LOR's in ein für die *Federation Engine* verständliches Format.

Die Transformation erfolgte durch die Bereitstellung von XSL-Stylesheets. Diese Stylesheets werden mittels Mappingregeln durch das Gateway erstellt. Diese Mappingregeln müssen für alle Systeme die das Gateway nutzen, angelegt werden, um die Erstellung der Stylesheets zu ermöglichen.

Innerhalb des Gateways existiert ein Cachespeicher, welcher verwendet wird, um einmal erzeugte Stylesheets mehrfach zu verwenden. Innerhalb dieser Arbeit führt das zu hohen Rechenzeiteinsparungen, da pro LOR nur einmal das Stylesheet erzeugt werden muss. Die Identifikation erfolgt durch eindeutige

Namensvergabe beim Abspeichern der Stylesheets im Cachespeicher. Als ID wird eine Kombination der beteiligten Systeme verwendet.

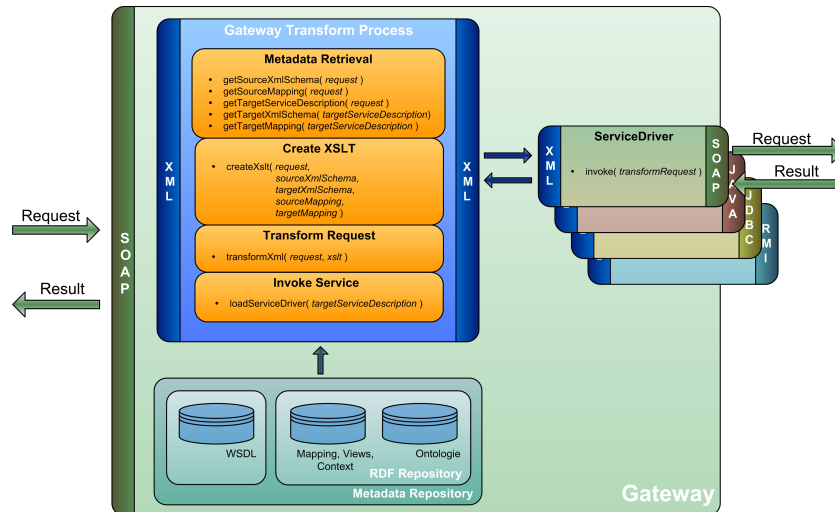


Abbildung 23: Funktionsweise des Gateways zur Transformation der Anfragen

Die Abbildung 23 stellt die verschiedenen Schritte des Gatewayprozesses dar. Diese werden nachfolgend kurz erläutert.

Im Funktionsbereich **Metadata Retrieval** werden die nötigen Daten für die Generierung des Stylesheets gesammelt. Hierzu zählen beispielsweise die Informationen, welche zur Auswahl der Mappingregeln benötigt werden.

Der **Create XSLT**-Bereich wird genutzt, um aufgrund der Daten, die im vorherigen Schritt gesammelt wurden, die Generierung des Transformationsstylesheets durchzuführen.

Innerhalb des **Transform Request**-Bereiches erfolgt die Übersetzung der Anfrage mittels des erzeugten Stylesheets.

Durch den **Invoke Service** erfolgt die Anfrage an das entsprechende System.

Die Antwort durchläuft die ersten drei Schritte des Gatewayprozesses und wird zurück an die *Federation Engine* gesendet.

Das Format der Mappingregeln, welche für die Erstellung der Stylesheets genutzt werden, soll nachfolgend erläutert werden.

Mappingregeln

Die Mappingregeln sind im RDF-Format erstellt worden, um die Semantik der Transformationen beschreiben zu können. Die Speicherung der Regeln erfolgt in einer RDF-Datenbank. Die verwendete Datenbank ist eine Sesame-Datenbank [Adu]. Der Vorteil bei der Nutzung einer RDF-Datenbank liegt in der Bereitstellung einer optimierten Anfragesprache, welches es erlaubt, direkt auf RDF-Elemente zuzugreifen.

Für jedes mögliche Element der Anfrage wird eine **Assoziation** erstellt. Innerhalb jeder **Assoziation** gibt es ein **Description**-Element, welches als Attribut den Namen der Mappingregel hat. Innerhalb dieses Elementes werden die Informationen gespeichert, welche nötig sind, um die Verbindung mit der Ontologie herstellen zu können. Zu jeder Mappingregel gehört eine eindeutige ID, welches mit dem Element **Id** festgelegt wird. Weiterhin gibt es jeweils drei Elemente für die Informationen über den Ort und den Typ des Elementes der Anfrage und für die Informationen des dazu passenden Ontologieelements.

Für die Anfrage sind es das Element **SchemaRef**, welches die Position des Elements im DOM-Baum der Anfrage präsentiert, das Element **SchemaValueTag**, welches die Position des Wertes in Abhängigkeit des Wertes im **SchemaRef**-Element enthält und das Element **SchemaValueType**, welches den Typ des Wertes angibt. Ein zusätzliches Element für die Anfrage ist das **ValueMap**, um konstante Werte beim Mapping hinzufügen zu können.

Die Elemente für die Ontologie haben analoge Bedeutungen. Die Namen lauten in der oben verwendeten Reihenfolge **OntologyRef**, **OntologyValueTag** und **OntologyValueType**. Ein optionales Element ist das Element **Condition**, welches Einschränkungen für die entsprechende Mappingregel festlegen kann. Diese können Einschränkungen des Wertebereichs, ebenso Bedingungen an andere Elemente der Anfrage, welche für die Mappingregel relevant sind, sein.

Die Elemente, welche die Typen enthalten, werden genutzt, um eventuell nötige Typumwandlungen vorzunehmen. Im Falle, dass ein Typ kein **SimpleType**, sondern ein **ComplexType** ist, wird eine Komposition bzw. eine Dekomposition durchgeführt.

Ein Beispiel für eine Mappingregel soll nachfolgend aufgezeigt werden.

```
<RDFNsId2:Association>
  <rdf:Description rdf:about="uri:LMSAnfrage#Title">
    <RDFNsId1:Id>7</RDFNsId1:Id>
    <RDFNsId1:SchemaRef>/request/title</RDFNsId1:SchemaRef>
    <RDFNsId1:SchemaValueTag>text()</RDFNsId1:SchemaValueTag>
    <RDFNsId1:SchemaValueType>xsd:string</RDFNsId1:SchemaValueType>
    <RDFNsId1:OntologyRef>/LORRequest/Title</RDFNsId1:OntologyRef>
    <RDFNsId1:OntologyValueTag>TitelText</RDFNsId1:OntologyValueTag>
    <RDFNsId1:OntologyValueType>xsd:string</RDFNsId1:OntologyValueType>
  </rdf:Description>
</RDFNsId2:Association>
<RDFNsId2:Association>
  <rdf:Description rdf:about="uri:LMSAnfrage#Title">
    <RDFNsId1:Id>7</RDFNsId1:Id>
    <RDFNsId1:SchemaRef>/SearchQuery/Title</RDFNsId1:SchemaRef>
    <RDFNsId1:SchemaValueTag>text()</RDFNsId1:SchemaValueTag>
    <RDFNsId1:SchemaValueType>xsd:string</RDFNsId1:SchemaValueType>
    <RDFNsId1:OntologyRef>/LORRequest/LearningObject/Title</RDFNsId1:OntologyRef>
    <RDFNsId1:OntologyValueTag>TitelText</RDFNsId1:OntologyValueTag>
    <RDFNsId1:OntologyValueType>xsd:string</RDFNsId1:OntologyValueType>
  </rdf:Description>
</RDFNsId2:Association>
```

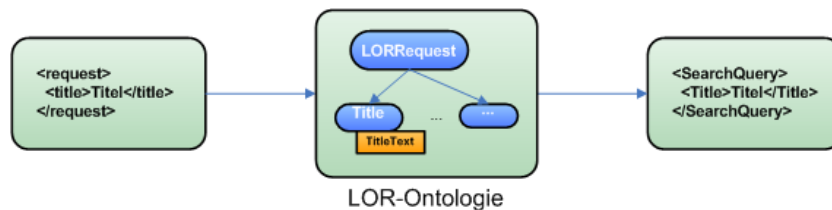


Abbildung 24: Beispiel des Mappings durch die LOR-Ontologie

Das Beispiel beschreibt die Übersetzung der Anfrage nach einem Titel durch die *Federation Engine* in die Anfrage an ein in der Arbeit eingesetztes LOR. Wie zu erkennen ist, wurde in beiden **Association**-Bereichen dieselben Ontologie-Definitionen verwendet.

Durch den Vergleich der Ontologeelemente werden die beiden passenden **Association**-Bereiche gesucht. Nachdem die entsprechenden **Association**-Bereiche gefunden wurden, werden die entsprechenden Stylesheets erzeugt.

Für das obige Beispiel würde das folgende Stylesheet erzeugt werden.

```
<xsl:stylesheet version="1.0" xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output encoding="ISO-8859-1"/>
  <xsl:template match="/">
    <SearchQuery>
      <xsl:apply-templates select="/request/title"/>
    </SearchQuery>
  </xsl:template>
  <xsl:template match="/request/title">
    <Title>
      <xsl:value-of select="text()"/>
    </Title>
  </xsl:template>
</xsl:stylesheet>
```

Neben der rein syntaktisch richtigen Transformation der Anfragen muss auch die Semantik berücksichtigt werden. Diese Problematik wird innerhalb des nächsten Abschnittes untersucht.

4.5.2. Semantische Transformation

Innerhalb des Gatewayansatzes[Sch03] werden gewisse semantische Transformationen betrachtet. Zu diesen zählen folgende:

- Komposition von Werten
- Dekomposition von Werten
- Transformation zwischen Datentypen und
- Abbildung von Konstanten.

Bei der **Komposition von Werten** werden die entsprechenden Werte mittels der XSLT-Funktion **concat()** miteinander verknüpft.

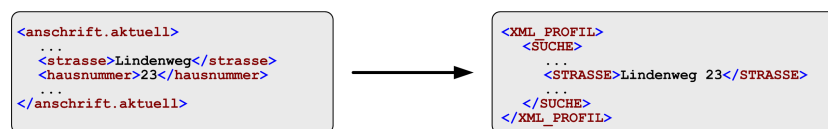


Abbildung 25: Beispiel für die Komposition von Werten[Sch03]

Die **Dekomposition von Werten** erfolgt mit Hilfe der XSLT-Funktion **tokenize()**, welche die einzelnen Bestandteile des entsprechenden Textknoten ermittelt. Diese werden intern in das entsprechende, durch die Mappingregeln vorgegebene, Format gebracht.

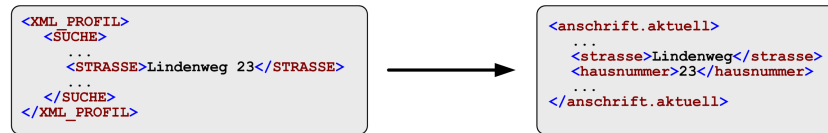


Abbildung 26: Beispiel für die Dekomposition von Werten[Sch03]

Transformationen zwischen Datentypen wird mittels verschiedener XSLT-Funktionen vorgenommen. Es ist aber durch die geringe Mächtigkeit der XSLT-Funktionen nicht möglich, alle Datentypen beliebig aufeinander abzubilden.

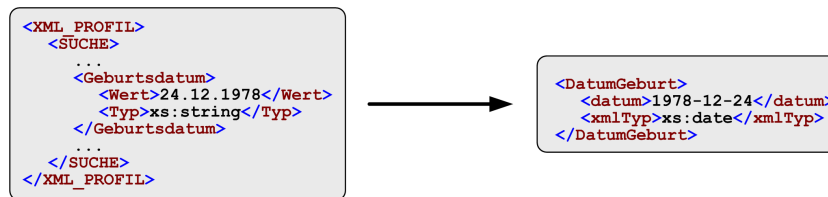


Abbildung 27: Beispiel für die Transformation zwischen Datentypen[Sch03]

Mittels der **Abbildung von Konstanten** wird es möglich, zusätzliche konstante Werte in die Anfragen einzubauen. Dies ist nötig, um Systeme zu unterstützen, die Schlüsselungen von bestimmten Werten nutzen.

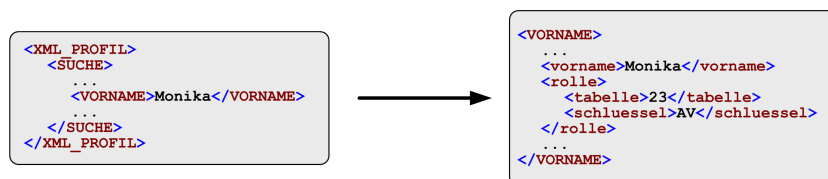


Abbildung 28: Beispiel für die Abbildung von Konstanten[Sch03]

Zusätzlich zu diesen semantischen Aspekten sollen in dieser Arbeit Betrachtungen über bool'sche Operationen durchgeführt werden. Diese Operationen sind wichtige Hilfsmittel, um Suchprozesse durch den Anwender besser steuern zu lassen.

Die in der Arbeit betrachteten Operationen sind:

- AND,
- OR und
- NOT.

Andere bool'sche Operationen werden nicht betrachtet, da sie in Suchmaschinen keine Rolle spielen bzw. durch die drei untersuchten Operationen abgebildet werden können.

Ein wichtiger Punkt um die Semantik der Anfragen berücksichtigen zu können, ist die Bereitstellung der Information, welche bool'schen Operationen durch das jeweilige LOR unterstützen. Weiterhin muss bekannt sein, welche Operation standardmäßig verwendet wird.

Durch gewisse Implementationen innerhalb der *Federation Engine* wäre es möglich, selbst nicht durch das LOR unterstützte Operationen durchzuführen. Für diesen Zweck sind mehrere Anfragen zu formulieren. Die Ergebnisse der einzelnen Anfragen müssen in einem Zusatzschritt dann richtig verknüpft werden.

Für eine Konjunktion ist eine selektive Verknüpfung der Ergebnisse nötig. In die Ergebnismenge werden nur Ergebnisse übernommen, die bei jeder Einzelanfrage in der Ergebnismenge vorkommen. Um die Disjunktion durchzuführen, werden die Ergebnismengen der Einzelanfragen additiv verknüpft. Soll eine Negation erfolgen, ist es nötig, die Schnittmenge der Ergebnismengen zu bilden.

Neben diesen generellen Betrachtungen, müssen auch hier Mischformen von bool'schen Operationen, sowie Vorrangregeln, z. B. durch Klammerung, berücksichtigt werden.

Innerhalb der Arbeit wurden diese Betrachtungen nicht weiter berücksichtigt, da die verwendeten LOR's nur konjunktive Verknüpfungen von Anfrageparameter unterstützen. Eine disjunktive Erweiterung der Anfrageparameter erfolgt nur innerhalb des Systems durch die Nutzung der Fachontologien.

Im nächsten Abschnitt wird die *Federation Engine* beschrieben. Sie nutzt die in diesem Abschnitt beschriebene *Transformation Engine*, in dem sie die Anfragen an die LOR's an dieses Engine weitergibt.

4.6. Federation Engine

Innerhalb dieses Abschnittes wird ein Konzept dargelegt, welches eine Föderation von Inhalten aus verschiedenen eLearning-System ermöglicht. Weiterhin wird betrachtet, wie zusätzliche Inhalte aus Wissensquellen in föderierte Kurse eingefügt werden können. Hierbei werden die in den vorangegangenen Abschnitten entwickelten Ansätze genutzt, um die auftretenden Probleme zu lösen.

Wichtige Teilaspekte, die hier besprochen werden sollen, sind die Erzeugung der Anfragen und das Zusammenfügen der gefundenen SCO's unter Berücksichtigung von didaktischen Konzepten.

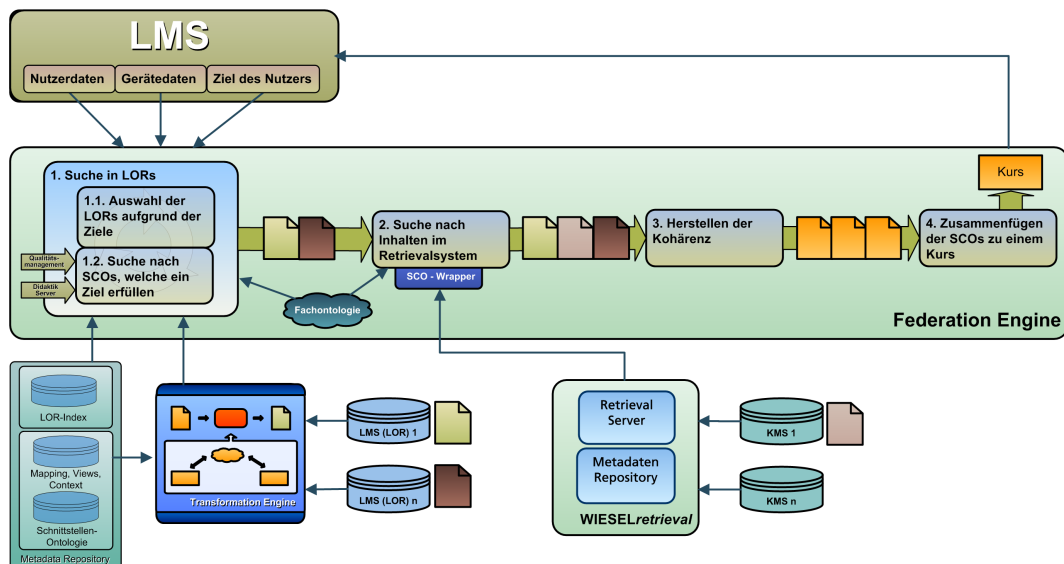


Abbildung 29: Grobalgorithmus für die Föderation

In der Abbildung 29 ist der gesamte Föderationsansatz zu sehen. Wie hier zu sehen ist, wurde der Föderationsansatz in vier Schritte aufgeteilt. Der erste Schritt ist die Suche in den LOR's. Nachdem durch die Suche ein Kurs entstanden ist, welcher nur aus Inhalten aus verschiedenen LOR's besteht, wird im zweiten Schritt eine Anreicherung dieses Kurses durch Nutzung von Daten aus anderen Wissensquellen vorgenommen. Der so entstandene erweiterte Kurs wird mittels spezieller Techniken auf ein einheitliches Layout gebracht. Im Anschluss wird dieser in ein Format gebracht, welches vom Lernmanagementsystem angezeigt werden kann.

4.6.1. Aspekte für die Föderation

Wie im Abschnitt 4.2.2 beschrieben wurde, haben Lernobjekten aus verschiedenen heterogenen LOR's meistens kein kohärentes Erscheinungsbild. Weiterhin erfolgt der didaktische Aufbau nicht nach einheitlichen Modellen. Um nach Abschluss der Föderation einen möglichst kohärenten Kurs zu haben, ist es nötig, Betrachtungen zur Herstellung der Kohärenz durchzuführen. Innerhalb der Arbeit werden aber nur Konzepte aufgezeigt, welche die Anpassung der visuellen Darstellung der SCO's ermöglichen. Konzepte für die Erstellung von Kohärenz im didaktischen Aufbau können bei [See03] nachgelesen werden.

Ein weiterer Aspekt ist die Berücksichtigung von Kosten für einzelne Lernobjekte. Durch die Erstellung von LOR's ist es nötig, ein Kostenmodell zu entwickeln, welches es ermöglicht, die in einem LOR enthaltenen Lernobjekte mit Preisen zu belegen. Innerhalb eines Föderationsansatzes muss gewährleistet sein, dass die entstehenden Kosten für den föderierten Kurs an die entsprechenden Instanzen weitergeleitet wird.

Unter dem Gesichtspunkt der kostenpflichtigen Lernobjekte gewinnt auch der Aspekt des Copyrights einen höheren Stellenwert. Um den Schutz des Copyrights der Lernobjekte gewährleisten zu können, müssen auch Techniken gefunden werden, die dieses ermöglichen. Mögliche Techniken sind die Verwendung von Wasserzeichen und Signaturen. Innerhalb der Arbeit soll aber nicht detaillierter auf diesen Aspekt eingegangen werden.

Durch die modulare Gestaltung der Arbeit und die offene Anfrageschnittstelle ist es aber möglich, in weiterführenden Arbeiten diese Aspekte in den entwickelten Ansatz zu integrieren.

Wie in diesem Abschnitt dargelegt, soll innerhalb der Arbeit keine Lösung spezieller Probleme der Föderation aufgezeigt werden, sondern ein Ansatz entwickelt werden, der die Erstellung eines Kurses aus Inhalten aus LOR's und Wissensquellen ermöglicht. Die Inhalte der in der Arbeit benutzten LOR's und Wissensquellen sind so gestaltet, dass die in diesem Abschnitt erläuterten Aspekte nicht berücksichtigt werden müssen.

4.6.2. Motivation des Suchansatzes

In der Regel wird man die Anforderungen, welche ein Nutzer an das LMS stellt, nicht mit einem vorgefertigten Kurs erfüllen können. Dies hängt damit zusammen, dass jeder Lernende ein anderes Vorwissen besitzt und somit für einen fertigen Kurs entweder zu viel oder zu wenig Wissen hat.

Dieses Problem soll mit diesem Föderationsansatz gelöst werden. Es ist somit notwendig, ausgehend vom aktuellen Wissen des Nutzers, die Ziele die der Nutzer vom LMS erwartet, zu erfüllen.

Es ist in den meisten Fällen nicht möglich, nur einen Weg von den Lernvoraussetzungen zu den gewünschten Zielen zu beschreiten, daher bietet sich ein Suchbaumansatz die Föderation an.

Die Ausarbeitung der verschiedenen Suchbaumansätze sind im Anhang zu finden. Aufgrund dieser Ausarbeitung wurde ein Suchalgorithmus ausgewählt, der sich für die Föderation am besten eignet.

4.6.3. Wahl eines Suchansatzes

Aufgrund der Anforderungen an die Arbeit bieten sich für die Föderation sowohl die Algorithmen der **Heuristisch Informierten Suche**, als auch die der **Optimalen Netzsuche** an.

Die Unterschiede liegen bei der Güte des föderierten Kurses. Die Algorithmen der **Heuristisch Informierten Suche** finden irgendeine Lösung für das Problem und sind daher in der Regel schneller als die Algorithmen der **Optimalen Netzsuche**, welche den optimalen Weg durch den Suchbaum finden.

In der Arbeit soll für die Suche ein **Best-First Search**-Algorithmus umgesetzt werden, da besonderer Wert auf einen möglichst kurzen Kurs gelegt wird.

4.6.4. Suche nach einer Heuristik

Eine wichtige Voraussetzung für einen effektiven Suchalgorithmus ist die Wahl einer guten Heuristik. Die Heuristik muss möglichst gute Ergebnisse liefern, wenn der richtige (optimale) Zweig des Suchbaumes betreten wurde und schlechte, wenn ein Zweig betreten wurde, der nicht zum Ziel führt.

Bei der Problemstellung der Arbeit bietet sich als Heuristik eine Kombination aus der Anzahl der schon benötigten SCO's und der Anzahl der noch nicht gelösten Zielen an. Ein weiterer Punkt, der in die Heuristik einfließt, ist die Berücksichtigung von Wissen, welches dem Lernenden vermittelt wird, welches aber nicht für die von ihm gestellten Anforderung notwendig ist. Zusätzliche Metadaten, wie die technischen Anforderungen, die Nutzerpräferenzen und didaktische Aspekte, fließen durch ein Ranking in die Heuristik ein.

Die Anzahl der SCO's soll möglichst klein gehalten werden, damit der Kurs möglichst schnell vom Lernenden erfasst werden kann. Dieser Punkt ist besonders bei der Bereitstellung von Kursen im Bereich WIESELMOBILE zu betrachten, da hier die Bandbreite ein ausschlaggebender Punkt für die Bereitstellung des Wissens ist. Für diesen Punkt muss aber die Größe der einzelnen SCO's mit berücksichtigt werden. Bestrebungen im Bereich der Erstellung von feingranularen SCO's [GAP02] würden diese Einflussgröße der Heuristik negativ beeinflussen. Um dies zu vermeiden, muss die richtige Balance zwischen der Menge der SCO's und dem zusätzlich erlernten Wissen gefunden werden. Durch die folgende Annahme sollte diese Balance gefunden werden.

Ein Kurs der sich aus wenigen großen SCO's zusammensetzt, liefert mehr zusätzliches Wissen, als ein Kurs, welcher aus mehreren kleinen SCO's besteht.

Die Anzahl der Ziele wird am Anfang der Suche stark zunehmen, da die gefundenen SCO's wieder bestimmte Voraussetzungen haben, die in der Regel noch nicht vorhanden sind. Ab einem bestimmten Punkt verringert sich die Anzahl der Ziele, da in vielen Fällen SCO's aus dem gleichen Fachgebiet einander ergänzen und somit einzelne Ziele gelöst werden, die nicht dem Hauptziel der gefundenen SCO entsprechen.

Neben diesen Werten wird ein Rankingwert berechnet, der den Kurs durch die Nutzer- und Gerätedaten an den Nutzer anpassen kann. Hierzu werden die Daten des Nutzers mit den Metadaten der SCO's abgeglichen. Falls Übereinstimmungen auftreten, fließen diese positiv in den Heuristikwert ein. Die verwendeten Daten für das Ranking beziehen sich z. B. auf das Format der Inhalte und den didaktischen Aufbau.

Damit ergibt sich als Heuristik folgende Formel:

$$\text{Heuristikwert} = f(S, Z, W, R)$$

S - Anzahl der bisher gefundene SCO's

Z - Anzahl der noch nicht erreichte Ziele

W - Anzahl des weiteren erlernten Wissens

R - Rankingeinfluss der Metadaten

Metadaten für die Heuristik

Um die Heuristik anwenden zu können, sind zwei Metadaten innerhalb der SCO's sehr wichtig. Das benötigte Vorwissen einer SCO fließt in die Menge der zu erreichenden Ziele ein. Durch dieses Metadatum wird die Menge der

Ziele vergrößert. Die Metadaten über die Ergebnisse einer SCO werden dafür genutzt, um die Menge der Ziele wieder zu verkleinern.

Zusätzlich kann jedes weitere Metadatum, welches in der SCO vorkommt, Einfluss auf den Heuristikwert haben. Dies erfolgt durch die Vorgaben des Ranking, in welchen die zu betrachteten Metadaten mit entsprechenden Gewichten definiert sind.

Im nächsten Abschnitt wird der eigentliche Suchalgorithmus beschrieben. Er benutzt den Suchansatz mit der vorgestellten Heuristik, um Anfragen zu generieren, die die Anforderungen des Nutzers an das LMS erfüllen.

4.6.5. Der Suchalgorithmus

Der Suchalgorithmus wird benutzt, um eine Suche nach SCO's in verschiedenen LOR's durchzuführen. Wie in Abbildung 29 ersichtlich wird, ist der Suchalgorithmus mit dem ersten Schritt identisch. Der Suchalgorithmus setzt sich aus zwei Teilschritten zusammen.

Im ersten Schritt werden die Oberklassenbeziehungen zum entsprechenden Begriff genutzt, um eine Auswahl der relevanten LOR's vorzunehmen. Dazu wird auf den LOR-Index zurückgegriffen. Der LOR-Index enthält Metadaten über die LOR's, welche eine Einordnung des LORs in eine Wissensdomäne ermöglicht. Die Metadaten werden in den LOR-Index eingetragen, wenn sich ein LOR am Föderationsystem anmeldet. Hierfür werden besonders die Hauptschlüsselworte der SCO's verwendet, da diese Auskunft über die Wissensdomäne des LORs geben.

Im zweiten Schritt erfolgt die eigentliche Suche nach den benötigten SCO's. Hierfür wird ausgehend vom Ziel, welches der Nutzer angegeben hat, der Suchbaum aufgebaut. Weiterhin fließen in die Suche, neben den rein inhaltlichen Aspekten, auch Aspekte der nutzbaren Technik, didaktische Aspekte und Vorlieben des Nutzers ein. Die benutzbare Technik ist besonders im Bereich von WIESELMOBILE wichtig. Die Nutzung von Metadaten über die technischen Voraussetzungen des Kurses werden daher im Abschnitt 4.7 genauer erläutert. Die didaktischen Aspekte, die in der Arbeit berücksichtigt werden, beziehen sich auf didaktische Makromodelle. Weitere Ausarbeitungen wurden schon im Abschnitt *Didaktische Konzepte* auf Seite 51 durchgeführt. Die Vorlieben des Nutzers werden sowohl im Abschnitt WIESELMOBILE, als auch im Abschnitt *Didaktische Konzepte* behandelt.

Innerhalb dieses Schrittes werden die in Abschnitt 4.2.1 beschriebenen Beziehungen innerhalb der Ontologien genutzt, um die Suchanfragen zu verbessern.

Synonymbeziehungen innerhalb der Ontologie werden genutzt, um die Anfrage zu erweitern. Dazu werden der Begriff und die dazugehörigen Synonyme innerhalb der Anfrage disjunktiv verknüpft.

Existieren für einen Begriff keine SCO's in den vorhandenen LOR's, wird durch die Nutzung von Unterklassenbeziehungen versucht, die entsprechenden Unterklassenbegriffe in den LOR's zu finden. Es wird in der Arbeit davon ausgegangen, dass die Summe des Wissens über die Unterklassenbegriffe einen ähnlichen Nutzen hat, wie das Wissen des Zielbegriffes.

Durch diese Vereinbarungen entstehen zwei Vektoren für jeden Suchknoten:

- Zielvektor und
- Vektor des Zusatzwissens

Die SCO's liefern neben der URI zu den SCO's auch die kompletten Metadaten. Die erhaltenen Metadaten fließen, zusammen mit den anderen Informationen, in den Heuristikwert ein.

Aufgrund des Heuristikwertes wird die beste SCO gewählt. Das Vorwissen, welches für die gefundene SCO noch benötigt wird, wird als Kindknoten eingefügt. Dies führt zur Erweiterung des Zielvektors im nächsten Durchlauf der Suche.

Führt ein Weg im Suchbaum in eine Sackgasse, wird mittels Backtracking zurück zum Vorgänger gegangen. Existieren für diesen Vorgängerknoten mehrere SCO's, wird die laut Heuristikwert nächstbeste SCO ausgewählt. Für diese SCO wird wieder das benötigte Vorwissen in den Zielvektor eingefügt.

Die beiden genannten Schritte werden solange durchlaufen, bis alle Ziele des Zielvektors erfüllt sind oder bis keine Verbesserung des Ergebnisvektors mehr erfolgt. Falls die Suche nicht für jeden Zielbegriff eine passende SCO findet, wird dies an das LMS weitergegeben und der Nutzer wird auf das fehlende Wissen hingewiesen.

Die gefundenen SCO's innerhalb des Suchbaumes werden nach Abschluss des Suchalgorithmus in die richtige Reihenfolge gebracht. Dazu werden die SCO beginnend bei den Blättern bis zur Wurzel hin, in einen Kursvektor aufgenommen. Nach dieser ersten Vorsortierung werden das Vorwissen und die Ergebnisse der SCO's untereinander verglichen. Treten hier Inkonsistenzen auf, werden diese durch Umsortierung behoben.

Nachdem die SCO's in die richtige Reihenfolge gebracht wurden, wird der gesamte Kurs durch Anfragen an das Retrievalsystem erweitert. Durch die

Course Enrichment Engine werden zu den einzelnen Kursteilen passende Informationen gesucht und zu SCO's verpackt.

4.6.6. Course Enrichment Engine

Die Anfrage nach zusätzlichen Informationen aus KMS erfolgt für jede SCO des Kursvektors. Um auch hier eine Einschränkung der KMS vor der eigentlichen Anfrage zu erreichen, wird auch hier eine Bestimmung aufgrund der Wissensdomäne vorgenommen. Im Anwendungsszenario existiert nur ein KMS, daher wurden hierfür keine weiteren Untersuchungen durchgeführt.

Die *Course Enrichment Engine* erweitert die Inhalte aus einem KMS so, dass es möglich ist, diese in einen LMS, welches sich an die SCORM-Spezifikation hält, darzustellen. Hier wird aufgrund von zusätzlichen Informationen, die im KMS gespeichert sind, eine Metadaten-Datei erstellt und diese zusammen mit dem Inhalt, welcher als XHTML-Seite gegeben ist, in eine SCO-Struktur verpackt. Um die Daten dynamisch zu halten, werden diese über ein eigendefiniertes Tag in die XHTML-Seite aufgenommen. Wird diese Dynamik nicht gewünscht, können die gegebenen Inhalte aber auch statisch in die XHTML-Seite aufgenommen werden.

Die Metadaten-Datei hält sich an den Metadaten-Standard des IEEE und ist somit auch SCORM-konform.

Die Engine dient weiterhin als Weiterleitungsstelle für die Anfragen des LMS an die KMS. Für die Arbeit wird vorausgesetzt, dass für jedes KMS ein eigener Wrapper programmiert wird. In weiteren Versionen, welche mit mehreren KMS zusammenarbeiten, müsste der Gateway-Ansatz, der für die Transformation der Anfragen an die verschiedenen LMS benutzt wird, auch für die Nutzung in diesem Kontext angepasst werden.

Das Verpacken der gefundenen Inhalte und den dazugehörigen Metadaten zu einer SCO erfolgt mittels des *SCOWrappers*. Der verwendete Algorithmus wird in Abbildung 30 dargestellt.

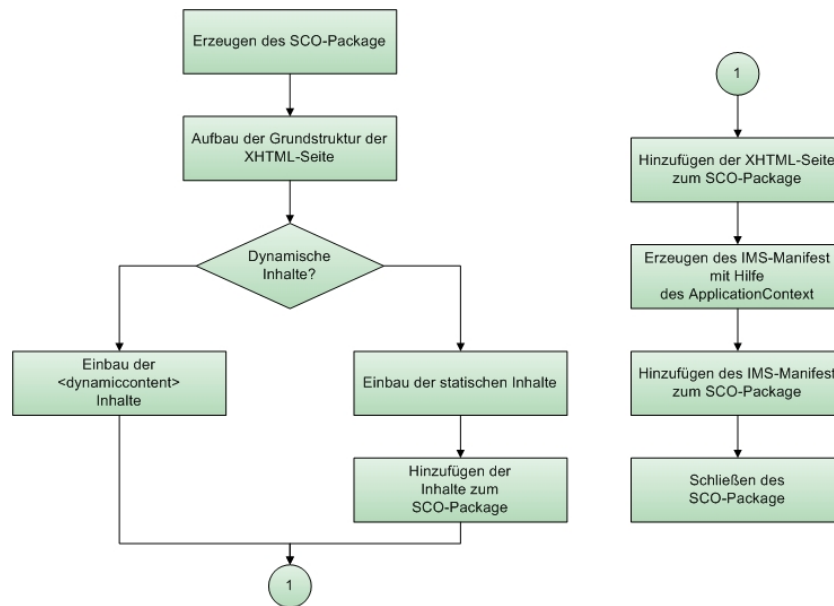


Abbildung 30: Algorithmus des SCOWrappers

Nachdem die SCO's in der richtigen Reihenfolge vorliegen und die Erweiterung durch die KMS in die Kursstruktur aufgenommen wurde, muss diese Kursstruktur zu einem SCORM-konformen Package verpackt werden, damit es im LMS dargestellt werden kann. Das Verpacken der SCO's wird im nächsten Abschnitt erläutert.

4.6.7. Erstellen eines Gesamtkurses

Im letzten Punkt der Föderation müssen die SCO's zu einem Gesamtkurs verknüpft werden. Im ersten Schritt wird eine Transformationspipeline abgearbeitet, um die Kohärenz innerhalb des Gesamtkurses zu erreichen. Diese Transformation bezieht sich aber nur auf das Aussehen der SCO's. In diesem Schritt werden nur die Farben und Schriften angepasst. Weitere Informationen über die Transformation des Layouts sind im Abschnitt WIESELMOBILE zu finden.

Im zweiten Schritt werden die SCO's in ein SCORM-Package verpackt und an das LMS weitergegeben. Das LMS ist für die Verwaltung des SCORM-Packages verantwortlich. Für das SCORM-Package muss zusätzlich eine Metadaten-Datei erzeugt werden, welche zusätzliche Informationen über den Kurs enthält und die Reihenfolge der einzelnen SCO's beschreibt.

Im letzten Abschnitt dieses Kapitels geht es um die Nutzung von Daten über den Nutzer und dessen Endgerät.

4.7. WIESELmobile

In diesem Abschnitt geht es um die Nutzung von mobilen Endgeräten im WIESELFRAMEWORK. Hierzu wird als Erstes betrachtet, wie die Daten des Gerätes gespeichert werden können. In einem nächsten Schritt werden Überlegungen angestellt, wie die Daten über den Nutzer und die Umgebung des Nutzers genutzt werden können, um innerhalb der Föderation bessere Kurse zu erhalten. Weiterhin wird eine Lösung aufgezeigt, wie SCO's an das jeweilige Endgerät angepasst werden können. In diesem Abschnitt werden auch die Transformationen zur Anpassung der Kohärenz besprochen.

Für die Erfassung der Gerätedaten wird CC/PP (Composite Capabilities/Preference Profiles) [Wor04a] benutzt. Neben den Gerätedaten können in diesem Format auch Daten des Nutzers gespeichert werden.

CC/PP

CC/PP ist eine W3C Recommendation. Sie wird von der "Device Independence WG" [Dev] benutzt und weiterentwickelt. RDF wird genutzt, um die Informationen über das Gerät und den Nutzer zu speichern. Die Daten können verwendet werden, um den Inhalt einer bestimmten Ressource an das jeweilige Gerät und den Nutzer anzupassen.

Mobile Konzepte der Arbeit

In der Arbeit wird CC/PP genutzt, um innerhalb der Föderation bestimmte zusätzliche Informationen in die Heuristik einfließen zu lassen. Unabhängig von der Wahl des Anzeigegerätes (mobile oder immobile) werden die Daten des Nutzers verwendet, um Vorwissen, didaktische Neigungen und andere personenbezogene Vorlieben für die Auswahl der SCO's zu nutzen.

Die Daten des Gerätes werden sowohl in der *Federation Engine*, als auch in der *Course Enrichment Engine* verwendet. Innerhalb der Föderation werden die Daten benutzt, um die Auswahl der SCO's zu verbessern. Die Verbesserung kommt durch den Einfluss der Gerätedaten auf die Heuristik zustande.

Sollten keine SCO's zur Verfügung stehen, die durch das Gerät unterstützt werden, müssen die nichtdarstellbaren Elemente entfernt bzw. transformiert werden. Für diese Transformation wird eine XSLT-Pipeline benutzt, die im nächsten Unterabschnitt beschrieben wird.

In der *Course Enrichment Engine* werden die Gerätedaten direkt für die Anfrage an das Retrievalsystem genutzt. Dies ist möglich, da die Informationen aus dem KMS als zusätzliches Wissen in die Kursstruktur eingebaut werden. Durch dieses zusätzliche Bereitstellen des Wissens können die Anfragen so gestellt werden, dass es auch möglich ist, keine SCO's zurückgeliefert zu bekommen. Durch die Gerätedaten kann bei Vorlage von mehreren Formaten für die gleichen Daten das günstigste Format gewählt werden.

Transformationspipeline

Die Transformationspipeline ist eine Kette von XSLT, welche nacheinander abgearbeitet werden. Sie wird benutzt, um die SCO's an das jeweilige Endgerät anzupassen. Eine weitere Aufgabe dieser XSLT-Kette ist die Herstellung der Kohärenz innerhalb des Gesamtkurses im Hinblick auf das Layout der einzelnen Kurseinheiten. Für die Anpassung der SCO's an das entsprechende Endgerät werden nichtunterstützte Teile der SCO entfernt.

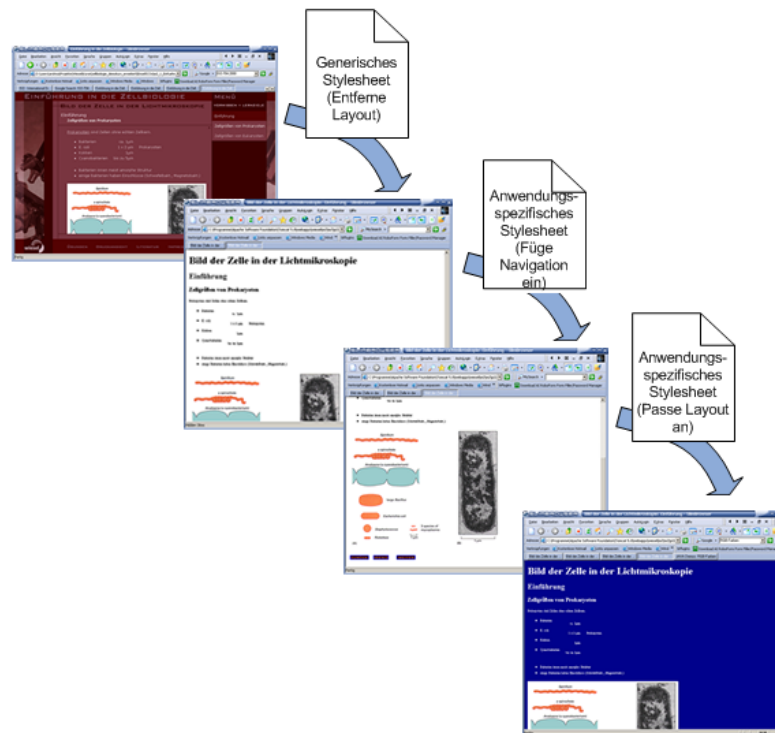


Abbildung 31: Beispiel für die Verwendung der Transformationspipeline

Ein Beispiel für die Benutzung der Stylesheet-Pipeline zeigt Abbildung 31. Hier wird ein SCO an ein mobiles Endgerät angepasst.

Die Erstellung der Pipeline erfolgt über Definitionen, welche auf dem Retrievalserver gespeichert sind. Innerhalb der Pipeline gibt es zwei verschiedene Arten von Stylesheets. Stylesheets der ersten Arten sind generisch und daher unabhängig von der Anwendung, welche auf das Retrievalsystem zugreift. Die zweite Art umfasst anwendungsspezifische Stylesheets. Sie sind abhängig von der Anwendung, welche auf die Informationen des Systems zugreifen will. Mithilfe dieser Gruppe wird die Transformation des Layout der einzelnen Kursteile durchgeführt.

Die Wahl, ob die Transformation im Retrievalsystem erfolgen soll oder ob die Pipeline zurückgegeben werden soll, kann über einen bool'schen Eingabeparameter bestimmt werden. In den ersten beiden Schritten des Algorithmus werden die Anwendung und das Profil für die Anfrage bestimmt. Im Anschluss werden für alle Komponenten des Profils und für alle Attribute der Komponenten die benötigten XSLT gesucht und an die Pipeline gehängt. Hierbei werden erst alle anwendungsspezifischen XSLT gesucht und danach alle generischen. Wenn die Pipeline vollständig ist, wird aufgrund des bool'schen Eingabeparameter entschieden, ob die Transformation durchgeführt wird oder ob die Pipeline nach außen weitergereicht wird.

Bestimmung der Gerätedaten

Momentan können die Daten der Geräte nicht direkt durch eine Anwendung erfragt werden. Daher ist es nötig, gewisse Informationen, welche durch die Geräte bereitgestellt werden, zu nutzen, um das CC/PP-Profil zu füllen. In der Arbeit werden Daten, welche mittels des HTTP-Protokolls versendet werden, hierfür genutzt. Innerhalb dieses Protokolls werden Informationen über die unterstützten MIME-Types, direkt oder indirekt, durch die Information der Browserversion, übermittelt. Weiterhin werden Daten über die vorhandene Software und deren Version bereitgestellt. Aufgrund dieser Daten werden einfache CC/PP-Profil erstellt und genutzt. Eine weitere Möglichkeit ist die Bereitstellung der Profile durch den Hersteller des Gerätes. Diese Profile können dann im Retrievalsystem gespeichert und über die IDs der Geräte benutzt werden. Wünschenswert wäre eine Speicherung der CC/PP-Profile direkt auf den Endgeräten und der Bereitstellung einer einheitlichen Schnittstelle zur Abfrage dieser Daten.

4.8. Zusammenfassung und Fazit

In diesem Kapitel wurden Ansätze aufgezeigt, wie die Anforderungen der Arbeit theoretisch umgesetzt werden können. Nachdem die Erweiterung der Ar-

chitektur vorgestellt wurde, ist auf die einzelnen Teilaufgaben, welche sich durch diese Architektur ergeben haben, eingegangen worden.

Im Abschnitt über WIESELFEDERATION wird auf die Transformation der Anfragen vom LMS an die verschiedenen LOR's eingegangen. Weiterhin wird die Föderation von SCO's aus den unterschiedlichen LOR's beschrieben. Innerhalb dieses Abschnitts werden der Suchalgorithmus und die Heuristik, welche zur Auswahl der SCO's genutzt wird, aufgeführt. Zusätzlich werden didaktische Konzepte untersucht, um die erhaltenen SCO's zu einem Gesamtkurs zusammensetzen zu können. Mittels einer Transformationspipeline werden die einzelnen SCO's vom Layout an das LMS angepasst, um eine möglichst kohärente Darstellung zu erhalten. Ein weiterer Punkt im Abschnitt WIESELFEDERATION ist die Einbindung von Wissen aus KMS. Hierzu wird ein Wrapper beschrieben, welcher Anfragen des LMS an den Retrievalserver übermittelt. Auch hier müssen Transformationen des Layouts vorgenommen werden, um diese in den Gesamtkurs aufnehmen zu können. Durch den Wrapper erfolgt auch die Generierung der SCO-Struktur, welche durch die Anreicherung der gefundenen Informationen durch Metadaten erreicht wird.

Der Abschnitt WIESELMOBILE erweitert die Transformationspipeline um die Transformationen, welche nicht darstellbare Elemente entfernen.

Ein weiterer Punkt der in diesem Kapitel abgehandelt wurde, ist die Verwendung von didaktischen Modellen für das Zusammenfügen der SCO's zu einem Gesamtkurs. Gleichzeitig wurde Wissen über die verwendeten didaktischen Konzepte zur Ermittlung der besten SCO mit Hilfe der Heuristik benutzt.

Bei der Bearbeitung der didaktischen Konzepte und bei der Untersuchung der bestehenden Metadatenspezifikationen wurden Probleme bei der Umsetzung erkannt. Die Probleme im Bereich der Metadatenspezifikation lagen in der ungenügenden Bereitstellung von benötigten Elementen. Durch eine Erweiterung des LOM-Standards konnten diese Probleme behoben werden. Als Erweiterung wurden in der Arbeit Elemente für die Aufnahme des Vorwissens, des Ergebnisses und der didaktischen Modelle bereitgestellt. Die Probleme innerhalb des Abschnitts *Didaktische Konzepte* konnten nur ungenügend gelöst werden. Das Hauptproblem ist die Umsetzung bestimmter didaktischer Modelle, um eine Nutzung innerhalb der Föderation zu ermöglichen. Ein zweiter Punkt sind die nicht vorhandenen Informationen über verwendete didaktische Konzepte innerhalb der SCO's.

Durch weitere Studien können die Probleme besser gelöst werden, da es Bestrebungen gibt, welche den didaktischen Aufbau von eLearning-Inhalten bewerten. Hierzu kann bei [din] nachgelesen werden. Dieser Vorschlag der DIN

PAS-Gruppe stellt eine Spezifikation bereit, welche einzelne Kurse mittels eines Punktsystems bewertet. Diese Spezifikation könnte benutzt werden, um den Einfluss der Didaktik auf die Heuristik besser abbilden zu können. Zum Zeitpunkt der Arbeit war noch keine konkrete Umsetzung dieser Spezifikation vorhanden, daher müssen weitere Untersuchungen zeigen, inwieweit diese Bewertungen nutzbar sind.

5. Prototyp und Implementation

Nach dem die theoretischen Grundlagen für die Arbeit im vorherigen Kapitel gelegt wurden, soll in diesem Kapitel auf die prototypische Umsetzung der Konzepte eingegangen werden. Die Umsetzung erfolgt im Rahmen des im Kapitel 1 beschriebenen Anwendungsszenario 1.2 auf Seite 2.

Der Prototyp erstellt aufgrund der Nutzervorgaben einen für diesen Nutzer spezifischen Kurs, welcher aus Inhalten aus heterogenen LMS und KMS besteht. Innerhalb dieses Kapitels wird auf die einzelnen Phasen der Erstellung von nutzerspezifischen Kursen eingegangen. Dies beginnt bei der Definition der Kursziele durch den Nutzer, über die Suche in den heterogenen LMS und KMS hin bis zum Zusammenbau der einzelnen SCO's zu einem Gesamtkurs. Im ersten Abschnitt wird auf die verwendeten Technologien eingegangen. Der zweite Abschnitt beschäftigt sich mit den verschiedenen Komponenten des Prototypen.

5.1. Verwendete Technologien

In der prototypischen Umsetzung der Arbeit wurden verschiedene Softwaremodule genutzt. Für die Umsetzung des Gateways, welcher die Anfragen des LMS übersetzt, damit diese für die heterogenen Service-Schnittstellen der LOR's genutzt werden können, wurde der Ansatz, welcher in der Diplomarbeit von Matthias Schulz [Sch03] erarbeitet wurde, modifiziert und eingesetzt. Die Beschreibung des Gateway-Ansatzes erfolgte im Kapitel 1.

Die einzelnen Softwarepakete sind:

- Sesame[Adu] - Sesame ist eine RDF-Datenbank. Sie wird im Prototypen genutzt, um die benötigten Ontologien zu speichern. Die Bereitstellung der Ontologien erfolgt über eine Web Service-Schnittstelle. Diese Web Service ist im WIESELRETRIEVAL (s. Kapitel 1) integriert. Mittels der Komponente *OntoResmo* des Retrieverservers können Anfragen an die Ontologie gestellt werden. Innerhalb der Service-Schnittstelle werden verschiedene Anfrage gekapselt und an die im Sesamesystem vorliegende Query-Engine weitergeleitet. Mittels dieser gekapselten Methoden können IDs, Sub- und Superklassen angefragt werden.
- Tomcat[The] - Tomcat ist eine Entwicklung der Apache Group und stellt einen Webserver auf der Basis von Java dar. Durch die Verwendung von JSP-Seiten, können dynamische Inhalte in die Webseiten integriert werden.

- Axis-Framework[Axi] - Das AXIS-Framework stellt eine Umsetzung des SOAP-Protokolls dar. Mittels des Frameworks können Beschreibungen vorgenommen werden, wie Daten mittels Web Services transportiert werden sollen.

Als Ontologien wurden folgende genutzt:

- Gene Ontology[Gen] - Die Ontologie enthält Zellfunktionen auf molekularer Ebene, wie katalytische Vorgänge und Transportvorgänge, biologische Prozesse, welche sich meistens aus mehreren Zellfunktionen zusammensetzen und die Komponenten, aus denen eine Zelle besteht.
- Cell Ontology[BA] - In dieser Ontologie sind die verschiedenen Zelltypen aufgeführt.

Der Prototyp wurde in Java [Sun] entwickelt, um plattformunabhängig zu sein. Ein weiterer Grund für die Wahl von Java ergab sich aus den verwendeten Vorarbeiten. Sowohl der Gatewayansatz, wie auch das WIESELRETRIEVAL wurden in Java implementiert. Die Verbindung der einzelnen Komponenten erfolgte mittels Web Services. Durch diese Web Services war es nicht nötig, direkten Zugriff mittels Java auf die Komponenten zu nehmen. Bestimmte Modifikationen waren innerhalb der Komponenten nötig, daher erschien die Wahl von Java sinnvoll.

Innerhalb des nächsten Abschnittes wird die Umsetzung der einzelnen Module des Prototyp besprochen. Dazu werden in den einzelnen Unterabschnitten die Anforderungen an die Module erläutert und die entwickelten Softwarelösungen besprochen.

5.2. Erweiterungen des LMS

Als Lernmanagementsystem wird COBILOT der Firma Anova verwendet. Für die Bereitstellung der Kurse im Internet wurde für das LMS ein Tomcat-Webserver [The] verwendet.

Um Kurse für jeden Nutzer individuell zu gestalten, ist es nötig, die Informationen über den jeweiligen Nutzer an die *Federation Engine* zu übergeben. Durch das Einloggen des Nutzers wird das entsprechende Nutzerprofil geladen. Dieses enthält Daten über den Nutzer. Hierzu zählen das Vorwissen (durch vorher besuchte Kurse), die Vorlieben und Informationen über das Endgerät, welches der Nutzer benutzt.



Abbildung 32: Webseite des LMS, in welchen die Suchanfrage eingegeben wird

Die Umsetzung der Suchmaske im LMS ist in Abbildung 32 dargestellt.

Um die Eingabe des gewünschten Kurszieles durch den Nutzer zu ermöglichen, wurde eine Eingabemaske in Form einer Webseite entwickelt. Diese wurde in das LMS integriert. In die Suchmaske wird das gewünschte Ziel des Kurses eingegeben. Eine Erweiterung des Webinterface unterstützt die Eingabe zusätzlicher Einschränkungen der Suchmenge. Die Suchanfrage wird als nächstes an die *Federation Engine* gesendet.

Die Anbindung des Modules WIESELFEDERATION erfolgt durch einen Web Service. Der Suchbegriff und die zusätzlichen Daten des Nutzers werden mittels XML-Syntax an die *Federation Engine* übergeben. Die Antwort der *Federation Engine* ist ein URI, welche den Ort des föderierten Kurses dem LMS bekannt gibt. Der Kurs wird vom LMS heruntergeladen und in die Verwaltungsstruktur aufgenommen. Hierdurch kann jetzt der Nutzer auf den föderierten Kurs zugreifen.

Zusätzlich zu dem föderierten Kurs werden die nicht erfüllbaren Voraussetzungen, die für das Verständnis des Kurses aber nötig sind, an das LMS übermittelt.

5.3. Föderation von Inhalten aus LOR's

In diesem Abschnitt werden die einzelnen Komponenten beschrieben, welche für die Suche von Inhalten in den verschiedenen LOR's nötig sind.

Innerhalb der *Federation Engine* wird als Erstes der Suchbaum generiert. Die Suchanfrage ist die Wurzel des Suchbaumes. Die Erstellung von Knoten erfolgt immer in der gleichen Reihenfolge.

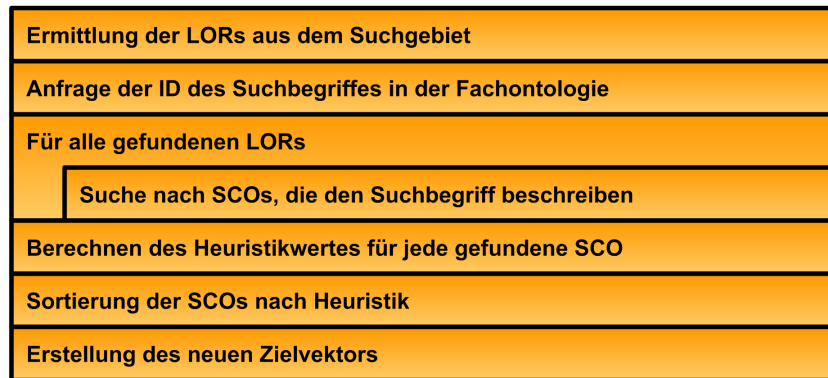


Abbildung 33: Erstellung von Knoten im Suchbaum

In der Abbildung 33 sind die verschiedenen Schritte der Erzeugung des Suchbaumes dargestellt. Der dargestellte Prozess wird solange durchgeführt, bis der Suchbegriff vollständig erklärt ist oder keine Verbesserung des Kurses erreicht werden kann.

5.3.1. Verwendung des LOR-Index

Wie in der Abbildung 33 dargestellt ist, erfolgt als Erstes die Bestimmung der LOR's, welche sich mit dem Fachgebiet, aus welchem der Zielbegriff kommt, beschäftigen.

Hierzu wird der LOR-Index benutzt. Dieser Index ist als RDF-Struktur erstellt worden und im Metadaten-Repository gespeichert. Er enthält Informationen über die verfügbaren LOR's, die an der Föderation teilnehmen können. Zu diesen Informationen gehören die WSDL-Beschreibung und die Metadaten, die das Fachgebiet beschreibt, über welche das LOR SCO's verwaltet. Der LOR-Index wurde angelegt, um die Performance der Anfragen zu verbessern, in dem die Anfragen nur an die LOR's gesendet werden, welche für den angegebenen Begriff relevant sind.

Um die LOR's zu identifizieren, wird der Zielbegriff mittels des LOR-Indexes überprüft. Werden keine LOR's gefunden, welche sich mit dem Zielbegriff beschäftigen, werden die Superklassen des Begriffes in der Fachontologie ermittelt. Mittels dieser Superklassen wird versucht, die Fachgebiete, welche für den Zielbegriff relevant sind, zu finden. Der Prozess der Superklassenermittlung erfolgt iterativ. Wird auf einer Stufe keine Übereinstimmung mit den Fachgebieten gefunden, wird die nächsthöhere Stufe untersucht. Das Abbruchkriterium ist erreicht, wenn mindestens ein LOR gefunden wurde oder eine vorgegebene Anzahl von Iterationen durchlaufen wurden.

5.3.2. Verwendung der Fachontologie

Im zweiten Schritt wird eine Anfrage an die Fachontologie gestellt, um eine eindeutige ID für den Suchbegriff zu ermitteln. Wird eine eindeutige ID gefunden, ersetzt diese den Suchbegriff. Dies ist nur nötig, wenn der Zielbegriff nicht als eindeutige ID aus einer Ontologie vorliegt.

Weiterhin wird innerhalb der Fachontologie nach Synonymen für den Begriff gesucht, um die Anfrage durch die gefundenen Synonyme zu erweitern. Diese Synonyme werden in der Anfrage disjunktiv zum Hauptsuchbegriff genutzt, um SCO's zu finden.

Ein weiterer Punkt in dem Fachontologien genutzt werden, ist die Nutzung von Subklassenbeziehungen. Dieses werden verwendet, wenn zu einem bestimmten Begriff keine SCO's innerhalb der LOR's gefunden wurde. Es wird davon ausgegangen, dass der Informationsgehalt, der durch die Bereitstellung von SCO's, welche die Subklassenbegriffe beschreiben, einen ähnlich großen Nutzen bieten, wie die Bereitstellung einer SCO, welche den Suchbegriff direkt als Kursziel hat. Die Verwendung der Subklassenbeziehungen erfolgt wie bei der Superklassenverwendung iterativ. Auch hier kann ein Level angegeben werden, bis zu welcher Ebene diese Auflösung in Subklassen erfolgen soll.

Nachfolgend soll ein Beispiel für die Nutzung der Fachontologie dargestellt werden.

Die Darstellung 34 zeigt verschiedene Synonyme für den Begriff Zellteilung, sowie verschiedene Subklassen zu diesem Begriff. Die anfängliche Suchanfrage lautet:

```
<request>
  <learningresult>cell division</learningresult>
</request>
```

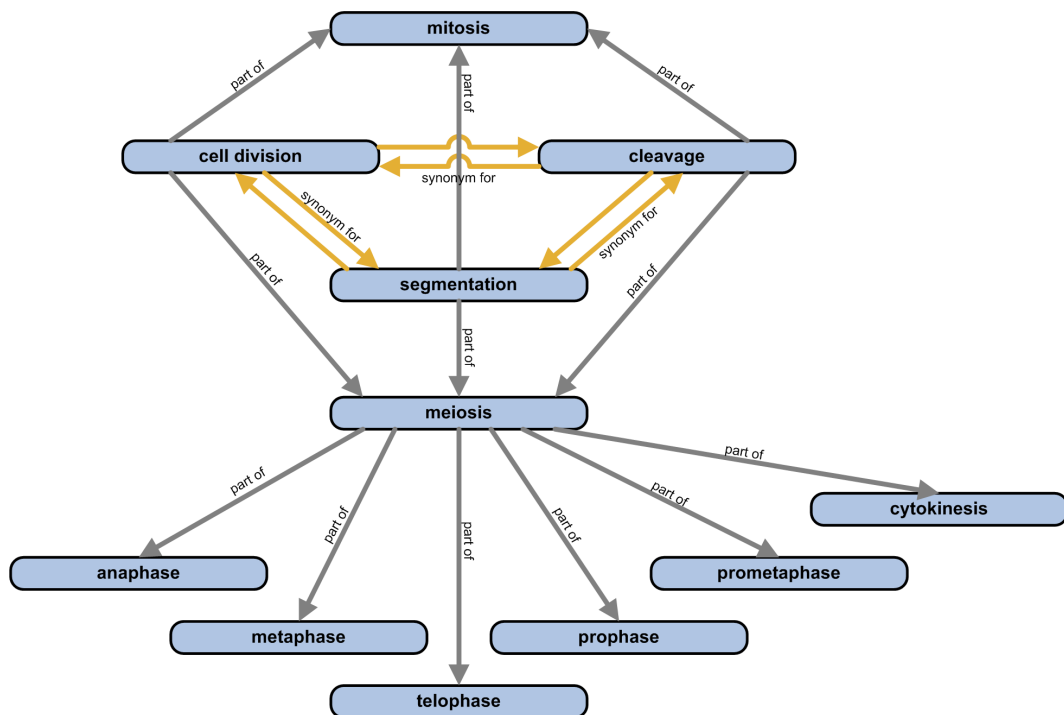


Abbildung 34: Beispiel für die Nutzung der Fachontologien

Durch die Ausnutzung der Synonyme würde im ersten Schritt die Anfrage wie folgt erweitert:

```

<request>
  <learningresult>cell division</learningresult>
  <learningresult>cleavage</learningresult>
  <learningresult>segmentation</learningresult>
</request>
  
```

Wenn für diese Suchanfrage keine Ergebnisse geliefert werden, würden die Subklassenbeziehungen genutzt werden, um eine Anfrage der Form:

```

<request>
  <learningresult>mitosis</learningresult>
</request>
<request>
  <learningresult>meiosis</learningresult>
</request>
  
```

zu erstellen. Wie zu erkennen ist, wird die einzelne Anfrage zu zwei Anfragen, um die Konjunktion der Ergebnismengen der beiden Begriffe zu erreichen. Wenn in diesem Fall nur SCO's, die als Kursziel den Begriff Mitose haben, gefunden wurden, werden die Subklassenbegriffe des Begriffes Meiose verwendet, wodurch folgende Anfrage aufgebaut wird:

```
<request>
  <learningresult>mitosis</learningresult>
</request>
<request>
  <learningresult>prophase</learningresult>
</request>
<request>
  <learningresult>prometaphase</learningresult>
</request>
<request>
  <learningresult>metaphase</learningresult>
</request>
<request>
  <learningresult>anaphase</learningresult>
</request>
<request>
  <learningresult>telophase</learningresult>
</request>
<request>
  <learningresult>cytokinesis</learningresult>
</request>
```

5.3.3. Gateway zur Anfrage an die LOR's

Im nächsten Schritt erfolgt die Suche innerhalb der ermittelten LOR's nach dem entsprechenden Suchbegriff und den vom Nutzer vorgegebenen Einschränkungen der Suchmenge. Dazu wird die Anfrage mittels XML gekapselt. Diese XML-Anfrage wird durch das Gateway in die für das jeweilige LOR benötigte Anfrage übersetzt. Danach wird es durch das Gateway an die LOR-Web Service-Schnittstelle gesendet. Das Ergebnis wird wieder durch das Gateway zurückübersetzt, damit es vom LMS verstanden wird. Hier erfolgt eine Auflösung der XML-Struktur. Die Antwort enthält jeweils:

- die URI der SCO, von welcher sie heruntergeladen werden kann und
- die kompletten Metadaten, welche die SCO beschreiben.

In den Metadaten sind die Begriffe enthalten, welche als Vorwissen für die SCO nötig sind. Diese werden für die Bildung des neuen Zielvektors benutzt.

Der Gateway führt folgende Schritte durch:

1. Ermittlung der Service-ID für die LOR aus dem LOR-Index
2. Suche der WSDL-Beschreibung aufgrund der Service-ID
3. Ermittlung der Mappingregeln für die Service-ID
4. Erstellung des Stylesheets zur Übersetzung der Anfrage
5. Übersetzung der Anfrage
6. Senden der Anfrage an den entsprechenden Web Service
7. Erstellung des Stylesheets zur Übersetzung der Antwort
8. Übersetzung der Antwort

Im LOR-Repository liegen für alle LOR's entsprechende WSDL-Beschreibungen. Diese enthalten neben den für den Web Service wichtigen Informationen auch Informationen über den Typ des Parameters. Mit der Service-ID, welche auch im LOR-Repository gespeichert ist, wird nach den entsprechenden Mappingregeln gesucht. Diese Mappingregeln enthalten Informationen über die verwendete Schnittstellenontologie. Durch diese Informationen kann ein Stylesheet erzeugt werden, welches die Anfragen des LMS in die Anfrage an das LOR übersetzt. Die übersetzte Anfrage wird per Web Service an das LOR gesendet. Nachdem vom LOR die Antwort eingetroffen ist, wird das entsprechende Stylesheet erzeugt, welches diese Antwort in eine für das LMS verständliche übersetzt. Die übersetzte Antwort wird jetzt vom LMS weiterbenutzt.

Es ist nicht nötig für jede Anfrage ein neues Stylesheet zu erzeugen, daher wurde im Gatewayansatz aus der Diplomarbeit von Matthias Schulz [Sch03] ein Cachingmechanismus entwickelt, welcher einmal erzeugte Stylesheets verwaltet. Durch die Service-ID erhält jedes Stylesheet einen eindeutigen Namen. Wird bei späteren Anfragen vom LMS das gleiche LOR angesprochen, wird das Stylesheet, welches vorher erzeugt wurde, wiederverwendet, wodurch die Performance des Gateway-Prozess erhöht wird.

5.3.4. Bestimmung der Relevanz

Nachdem die Metadaten aufgelöst wurden, wird für jede SCO die Relevanz mittels der Heuristik ermittelt.

In die Heuristik fließen ein:

- die Anzahl der für die SCO benötigten Voraussetzungen,
- die Anzahl der durch die SCO erfüllten Voraussetzungen,
- die momentane Tiefe des Suchbaumes und
- die Vorgaben des Nutzers (didaktische Vorlieben, gewünschte Medientypen, ...).

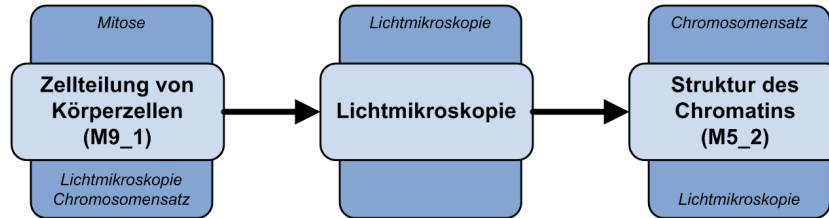
Der neue Zielvektor setzt sich aus den Zielen des alten Suchvektors und den Begriffen, die als Voraussetzungen für die SCO nötig sind, zusammen. Abgezogen von dieser Menge von Zielen werden die Ziele, die schon durch eine andere SCO erklärt wurden. Weiterhin werden die Ziele entfernt, die durch das Wissen des Nutzers schon vorhanden sind.

Sind in einem Knoten des Suchbaumes mehrere SCO's vorhanden, kann durch Backtracking innerhalb des Baumes die Suche verbessert werden. Die Suche erfolgt so lange bis alle benötigten Begriffe durch SCO's erfüllt sind oder bis keine weiteren SCO's gefunden werden. Wenn nicht alle Ziele erreicht werden konnten, erfolgt eine Rückmeldung des noch benötigten Wissens an den Nutzer. Diese Informationen über das benötigte Wissen wird als Webseite im LMS dargestellt.

5.3.5. Sortierung der gefundenen SCO's

Nachdem der bestmögliche Kurs gefunden wurde, wird der Vektor der SCO umgekehrt und nochmal auf die richtige Reihenfolge überprüft. Dies ist nötig, da der Suchalgorithmus vom Ziel ausgeht und das benötigte Vorwissen immer weiter verfeinert. Durch diese Vorgehensweise kann es vor der Sortierung zu Verschiebungen von SCO's kommen, die für mehrere SCO's als Vorwissen dienen. Zur Erläuterung dient das nächste Beispiel.

Kursvektor vor der Sortierung



Kursvektor nach der Sortierung

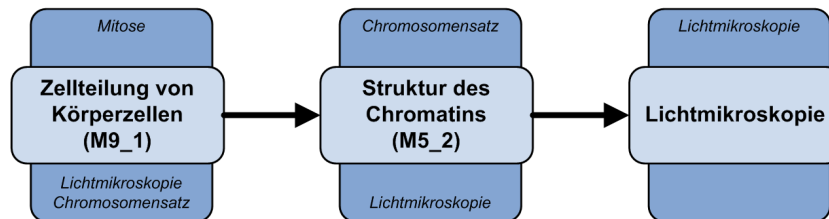


Abbildung 35: Beispiel für die Sortierung der gefundenen SCO's

Wie in Abbildung 35 zu sehen ist, benötigen die SCO's M9_1 und M5_2 als Voraussetzung Wissen über Lichtmikroskopie. Da M9_1 näher an der Wurzel des Suchbaumes liegt, wird die SCO, welche Lichtmikroskopie erklärt, direkt vor dieser und hinter der SCO M5_2 angeordnet.

Dieses Beispiel zeigt deutlich, warum eine Anpassung der Reihenfolge erfolgen muss.

Nachdem der Kurs ermittelt und in die richtige Reihenfolge gebracht wurde, wird dieser Kurs um Inhalte aus verschiedenen KMS ergänzt.

5.4. Einfügen von Inhalten aus KMSs

Die *Course Enrichment Engine* generiert aus den Anfragen der *Federation Engine* neue Anfragen, welche an die verschiedenen Wissensmanagementsystemen übermittelt werden. Die Anfragen der *Federation Engine* bestehen aus den Lernzielen der SCO's, welche mittels des föderierten Kurses vermittelt werden. Diese werden an das Modul REPOSITORYINFO des Retrievalservers übermittelt.

Im ersten Schritt erfolgt, wie bei der Suche in den LOR's, eine Auswahl von KMS des entsprechenden Fachgebietes. Hierfür wird derselbe Algorithmus ver-

wendet, wie im Fall der LOR's. Schrittweise werden die Superklassen der Begriffe der SCO's gesucht. Die Superklassen werden zur Auswahl der KMS verwendet. Die schrittweise Vorgehensweise erfolgt solange, bis mindestens ein KMS gefunden wurde, welches dem entsprechenden Fachgebiet zugeordnet ist. Für die Anfrage der Superklassenbeziehungen, wird das Modul *OntoResom* des Retrievalservers verwendet, welches gekapselte Anfragen an das Ontologie-Repository mittels einer Web Service-Schnittstelle bereitstellt.

In den gefundenen KMS erfolgt jetzt die Suche nach Inhalten zu den Lernzielen der zugeordneten SCO. Werden Inhalte gefunden, werden die Anfragen an das Modul *RepositoryInfo* mittels des Tags `<dynamiccontent>` in eine XHTML-Webseite integriert. Durch die Verwendung dieses Tags ist es dem LMS möglich, bei jedem Aufruf der Webseite eine Anfrage an den Retrievalserver zu stellen. Durch diese Verfahrensweise erfolgt eine dynamische Erweiterung des Kurses durch die Inhalte der verschiedenen KMS.

Für die Anfrage an die KMS werden neben den Suchbegriffen auch Informationen über den Nutzer und dem verwendeten Gerät übermittelt. Durch diese zusätzlichen Informationen werden die Daten in der richtigen Qualität und im richtigen Format geliefert. Dies spielt besonders im Kontext der mobilen Endgeräte eine wichtige Rolle.

Bemerkung: Das `<dynamiccontent>`-Tag wurde im WIESEL-Projekt entwickelt, um dynamische Inhalte in Webseiten integrieren zu können. Es wird vom LMS zur Laufzeit aufgelöst.

Die so erzeugte Webseite wird mit Metadaten, die vom KMS bereitgestellt werden, erweitert und zu einer SCO verpackt. Diese SCO wird in die Kursstruktur hinter der jeweiligen SCO eingeordnet.

Nachdem sowohl Wissensinhalte aus den verschiedenen LMS, als auch aus den zur Verfügung stehenden KMS in die Kursstruktur eingeordnet wurden, ist es noch nötig, das Aussehen der Kursteile zu vereinheitlichen, um für den Nutzer ein kohärenten Kurs bereitzustellen.

5.5. Erzeugung von Kohärenz

Die Herstellung der Kohärenz im Layout der SCO's erfolgt über eine Transformationspipeline, welche sich aus verschiedenen XSL-Stylesheets zusammensetzt. Die Pipeline baut sich aus zwei Arten von Stylesheets auf. Zu unterscheiden sind:

- die generischen Stylesheets und
- die anwendungsspezifischen Stylesheets.

In der Pipeline werden als Erstes die generischen Stylesheets angeordnet. Mittels dieser werden beispielsweise die Layoutdefinition aus den SCO's entfernt. Dazu werden:

1. externe CSS-Verweise entfernt,
2. interne CSS-Definitionen gelöscht,
3. Farben (Hintergrund, Schrift) zurückgesetzt und
4. Schriftformatierungen entfernt.

Die nachfolgenden anwendungsspezifischen Stylesheets dienen dazu, Anpassungen des Layouts für die jeweilige Anwendung vorzunehmen. Die Definition des Layout kann entweder durch die Beschreibung des Layouts in einer CSS-Datei oder direkt innerhalb der Webseite erfolgen.

5.6. Erstellen des Kursdatei

Der letzte Schritt innerhalb der *Federation Engine* ist das Verpacken der SCO's zu einem SCORM-konformen Kurs. Dazu werden die Metadaten der einzelnen SCO in die Metadaten-Datei des Kurses integriert. Weiterhin wird in der Metadaten-Datei die Reihenfolge der SCO beschrieben.

Der so erhaltene Kurs wird an das LMS gesendet und hier gespeichert. Der Nutzer, der die Anfrage nach dem Kurs gestellt hat, kann jetzt auf den Kurs zugreifen. Im Falle, dass nicht alle benötigten Begriffe gefunden wurden, wird der Nutzer darauf hingewiesen und aufgefordert dieses Wissen auf anderen Weg zu erlangen. Hat der Nutzer den kompletten Kurs absolviert, wird dieser gelöscht.

5.7. Zusammenfassung und Fazit

In diesem Kapitel wurde auf die Implementation des Prototypen eingegangen. Es wurde dargelegt, wie die verschiedenen Komponenten des Systems durch Web Services miteinander verbunden wurden.

Ein wichtiger Punkt bei der Implementation war die Einbindung von bestehenden Softwarelösungen in das System. Diese Einbindung erfolgt über die von den Systemen bereitgestellten Web Service-Schnittstellen.

Durch die Implementation des Prototypen konnte der entwickelte Suchalgorithmus getestet werden. Die Ergebnisse der Testanfragen erfüllten die gestellten Anforderungen, in dem der Algorithmus Kurse föderierte, welche in vorher definierten Suchszenarien beschrieben wurden.

Die im Lösungsansatz entwickelte Architektur des Modules WIESELFEDERATION stellte sich als günstig, für die Erfüllung der Anforderungen, heraus und wurde im Prototypen umgesetzt.

Die anschließenden Test zeigten gute Ergebnisse. Weitere Tests mit größeren Datenmengen müssen zeigen, ob der Ansatz auch in diesen größeren Datenmengen gute Kurse föderiert und wie stark sich die Menge an LOR's auf die Geschwindigkeit der Föderation auswirkt.

Die in der Arbeit verwendeten Ontologien erfüllten die geforderten Voraussetzungen. Durch die Verwendung der Beziehungen der Klassen untereinander konnten die Auswahl von Lernobjekten aus den LOR's verbessert werden. Wichtig in diesem Bereich wäre eine Standardisierung der Ontologien, wo-durch eine eindeutige Identifizierung von Begriffen durch Ontologie-IDs möglich wird.

Weiterhin wurde während der Implementation festgestellt, dass es nötig ist weiterführende Betrachtungen durchzuführen, wie die Semantik besser innerhalb von Web Services beschrieben werden kann.

6. Zusammenfassung und Ausblick

6.1. Zusammenfassung

Die Arbeit beschäftigt sich mit dem Thema der Föderation von Inhalten aus verschiedenen Lernmanagementsystemen und der Integration von zusätzlichen Informationen aus Wissensmanagementsystemen. Es wurde ein möglicher Weg aufgezeigt, um dieses Ziel zu erreichen. Semantische Dienste wurden verwendet, um die Nutzung der verschiedenen Systemschnittstellen zu ermöglichen. Weiterhin wurden Techniken des Semantic Web für die Verbesserung der Anfragen an die Systeme genutzt.

Die Anbindung der einzelnen Teilbereiche wurde so umgesetzt, dass die einzelnen Module möglichst unabhängig von einander sind. Hierzu wurde die Technik der Web Services angewendet. Im Prototypen wurde die Anfrage durch das LMS an die *Federation Engine* als Web Service gestaltet. Weiterhin müssen alle Systeme, LMS und KMS, welche an der Föderation teilnehmen wollen, eine Web Service-Schnittstelle bereitstellen, die die Suchanfrage entgegennimmt und die entsprechende Antwort zurückliefert. Für die Übersetzung und Übermittlung der Anfragen wurde eine Gatewayansatz [Sch03] verwendet. Dieser Gatewayansatz benutzt um semantische Konstrukte erweiterte WSDL-Beschreibungen, um die Anfrage zu transformieren.

Durch die Nutzung von Fachontologien wurden die Anfragen der *Federation Engine* erweitert. Diese Erweiterung wurde durch die Beziehungen der Klassen in der Ontologie untereinander erreicht. Besonders wichtig in diesem Zusammenhang waren Synonymbeziehungen. Zusätzlich wurden Subklassenbeziehungen genutzt, um SCO's zu nicht vorhandenen Begriffen zu finden.

In der Arbeit mussten die Metadatendefinition des IEEE angepasst werden, um die Föderation überhaupt möglich zu machen. Die Erweiterungen betrafen die Aufnahme des benötigten Vorwissens und die Ergebnisse, welche durch die entsprechende SCO vermittelt werden. Weiterhin wurde ein Metadatum eingeführt, welches Informationen über das verwendete didaktische Modell enthält. Diese Erweiterungen wurden intern für das Projekt getroffen und kommen in der Praxis nicht vor. Daher ist es schwierig, andere LOR's momentan an die Föderation anzuschließen.

Neben der Föderation von Inhalten aus verschiedenen LMS wurden auch Möglichkeiten untersucht, wie Inhalte aus Wissensmanagementsystemen in föderierte Kurse aus eLearning-Inhalten einfließen können. Hierbei ging es nicht um die Nutzung von Wissen aus KMS als Ersatz für Inhalte aus LMS, sondern

um die Erweiterung des durch die SCO's bereitgestellten Wissen durch Beispiele und zusätzliche Materialien. Hierbei wurde untersucht, wie dieses Wissen dem Nutzer zur Verfügung gestellt wird. Es wurden zwei Möglichkeiten der Darstellung in der Arbeit diskutiert. Auf der einen Seite können diese Inhalte als statische HTML-Seiten bereitgestellt werden. Die andere Möglichkeit ist die Bereitstellung von HTML-Seiten mit dynamischen Inhalten. Im Prototyp wurde die zweite Möglichkeit umgesetzt, da durch das Lernmanagementsystem schon eine Auflösung von dynamischen Inhalten vorgesehen ist.

Ein weiterer Punkt in der Arbeit war die Untersuchung von Technologien, mit denen eine geräte- und situationsabhängige Ermittlung von Kursen und Kursteilen möglich ist. Hierzu wurde das CC/PP-Protokoll [Wor04a] genauer betrachtet. Neben der theoretischen Betrachtung der Nutzbarkeit dieser Technologie für die Föderation wurden erste Versuche im Prototypen gemacht, um verschiedene Endgeräte zu berücksichtigen. Hierzu wurden technische Einschränkungen in die Suche nach den SCO's eingebaut. Hierdurch werden nur Inhalte gefunden, die den Anforderungen an die Formate genügen.

Es wurden neben der rein inhaltlichen Föderation auch Ansätze untersucht, wie didaktische Modelle berücksichtigt werden können. Weiterhin wurden Untersuchungen vorgenommen, ob es Bestrebungen gibt, welche sich mit der Definition und Bewertung der Qualität von Lerneinheiten beschäftigen. In beiden Bereichen konnten erste Ansätze aufgezeigt werden. Die didaktischen Modelle wurden berücksichtigt, um SCO's zu suchen, welche sich besonders gut für ein spezielles Modell eignen.

Durch die Verwendung verschiedener eLearning-Systeme und Wissensquellen entsteht bei der Föderation ein Kurs, welcher möglicherweise nicht kohärent ist. In der Arbeit wurde ein Ansatz beschrieben und umgesetzt, wie die Kohärenz auf der Ebene der Präsentation der Inhalte erreicht werden kann. Hierzu wird eine Transformationpipeline genutzt. Die verwendeten Stylesheets der Pipeline lassen sich in zwei Kategorien unterteilen. In der ersten Kategorie sind generische Stylesheets enthalten, die zum Entfernen von Layoutvorschriften innerhalb der SCO's dienen. Die zweite Kategorie umfasst anwendungsspezifische Stylesheets, welche abhängig von dem System sind, welches diese Stylesheets benutzt. In den Stylesheets sind die Layoutvorschriften des Systems definiert. Diese Layoutvorschriften werden auf das SCO, welche durch die generischen Stylesheets bereinigt wurde, angewendet.

6.2. Ausblick

Der in der Arbeit umgesetzte Ansatz ist nur prototypisch, daher gibt es verschiedene Ansatzpunkte, bei denen Verbesserungen und Erweiterungen denkbar sind. Weiterhin sind einige der verwendeten Techniken auch nur prototypisch vorhanden, wodurch auch hier Bedarf besteht, die Verbesserung der Techniken in zukünftigen Arbeiten zu berücksichtigen.

Ein wichtiger Punkt, der in nachfolgenden Arbeiten untersucht werden muss, ist die durch Standardisierungen entstehenden Erweiterungen der Metadaten. Innerhalb der Arbeit wurde festgestellt, dass es durch den zur Verfügung stehenden Metadaten-Standard des IEEE nicht möglich ist, eine Föderation von Inhalten aus LOR's durchzuführen. Die Arbeit hat gezeigt, dass Beschreibungen der Voraussetzungen für eine SCO genauso vorhanden sein müssen, wie eine Beschreibung der Ergebnisse der SCO.

Ebenso müssen weitere Entwicklungen im Bereich der Einbettung von Semantik in Web Service betrachtet werden. Ein Framework, welches sich mit der Einbettung von Semantik in Web Services beschäftigt ist OWL-S [MBD]. Für die Semantikbeschreibung mittels Ontologien wäre es günstig, wenn diese Ontologien durch Standardisierungen vereinheitlicht werden. Weiterhin existieren in den meisten Bereichen keine Ontologien, wodurch in diesen Bereichen der Föderationsalgorithmus nicht angewendet werden kann.

Zusätzlich zu den in der Arbeit verwendeten Techniken zur Herstellung der Kohärenz des Layouts der SCO's, können weitere Arbeiten Ansätze suchen, wie auch eine Kohärenz auf der Ebene des inhaltlichen und didaktischen Aufbaus erreicht werden kann. Ansätze, welche sich mit diesem Thema beschäftigen sind [See03].

In weiteren Arbeiten müssen auch die rechtlichen Aspekte näher berücksichtigt werden. Weiterhin können Ansätze untersucht werden, wie kostenpflichtige SCO's in das Föderationssystem aufgenommen werden können.

Weiterhin von Interesse für die Föderation ist die qualitative Bewertung der SCO's. Hiermit können SCO's einer bestimmten Qualitätsstufe ermittelt und föderiert werden. Dieser Prozess der Bewertung von SCO ist auch für die Realisierung von Kostenmodellen nötig, da hiermit Preis-Leistung-Bewertungen ermöglicht werden. Hierfür müssen die Entwicklungen der DIN-PAS [din] weiterverfolgt werden.

A. Anhang

A.1. SMETE Sucheergebnis

```
<schema targetNamespace="urn:SmeteSearch"
  xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
  <complexType name="SmeteSearchResult">
    <sequence>
      <element name="ListRecords"
        nillable="true"
        type="impl:ListRecordsType"/>
    </sequence>
  </complexType>
  <complexType name="ListRecordsType">
    <sequence>
      <element name="startIndex"
        type="xsd:int"/>
      <element name="endIndex"
        type="xsd:int"/>
      <element name="totalResultsCount"
        type="xsd:int"/>
      <element maxOccurs="unbounded"
        name="error"
        nillable="true"
        type="impl:errorType"/>
      <element maxOccurs="unbounded"
        name="record"
        nillable="true"
        type="impl:recordType"/>
    </sequence>
  </complexType>
  <complexType name="errorType">
    <sequence>
      <element name="code"
        nillable="true"
        type="impl:errorcodeType"/>
    </sequence>
  </complexType>
  <simpleType name="errorcodeType">
```

```
<restriction base="xsd:string">
  <enumeration value="notAllowed"/>
  <enumeration value="noRecordsMatch"/>
  <enumeration value="badQuery"/>
</restriction>
</simpleType>
<complexType name="recordType">
  <sequence>
    <element maxOccurs="1"
      minOccurs="0"
      name="metadata"
      nillable="true"
      type="impl:metadataType"/>
    <element maxOccurs="1"
      minOccurs="0"
      name="score"
      type="xsd:int"/>
  </sequence>
</complexType>
<complexType name="metadataType">
  <sequence>
    <element name="value"
      nillable="true"
      type="xsd:string"/>
  </sequence>
</complexType>
<element name="SmeteSearchResult"
  nillable="true"
  type="impl:SmeteSearchResult"/>
</schema>
```


A.2. Suchalgorithmen

Die Informationen für diesen Abschnitt wurden aus der Webseite [suc] extrahiert. Hier eine Aufstellung der untersuchten Algorithmen:

- Blinde Suche
 - Tiefensuche (depth-first search)
 - Breitensuche (breadth-first search)
 - Nichtdeterministische Suche
- Heuristisch informierte Suche
 - Hill-Climbing
 - Beam Search
 - Best-First Search
- Optimale Netzsuche
 - British Museum Procedure
 - Branch-and-Bound Search (BaB Search)
 - BaB Search mit Abschätzung der Restkosten
 - BaB Search mit Eliminierung von langen Doppelwegen und
 - A* Algorithmus.

Blinde Suche

Die Algorithmen der **Blinden Suche** nutzen für die Suche keine zusätzlichen Informationen, um eine höhere Effektivität zu erreichen. Durch diese Einfachheit der Algorithmen sind sie nur für bestimmte Arten von Problemen nutzbar.

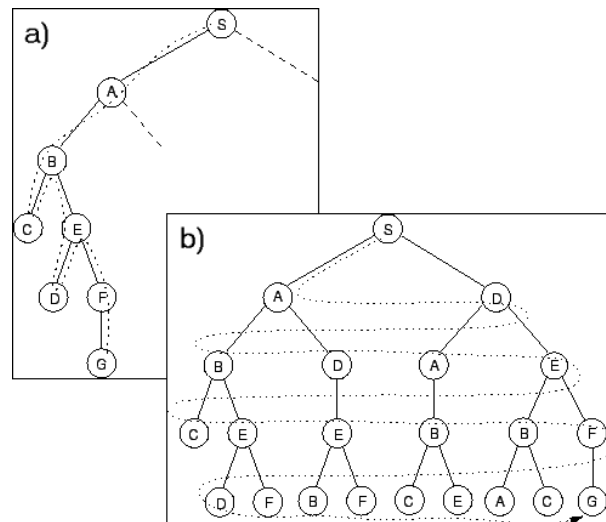


Abbildung 36: Algorithmen der Blinden Suche a) Tiefensuche und b) Breiten-suche

In der **Tiefensuche** wird vom Startknoten ausgehend der erste Kindknoten besucht. Falls dieser Kindknoten nicht das Ziel ist, wird ausgehend von diesem Knoten, wieder der erste Kindknoten besucht. Dieses Absteigen in die Tiefe des Suchbaumes wird solange fortgesetzt, bis entweder das Ziel gefunden ist, eine Schleife betreten wurde oder der Kindknoten keine weiteren Kinder hat. In den letzten beiden Fällen wird der Baum rückwärts durchlaufen bis ein Knoten erreicht wird, der noch nicht besuchte Kindknoten hat. Diese werden im nächsten Schritt mittels des Algorithmus durchsucht. Der Algorithmus bricht ab, wenn das Ziel erreicht ist. Durch diese Vorgehensweise eignet sich der Algorithmus nur für Suchbäume, deren Äste nur eine vertretbare Länge haben.

Bei der **Breitensuche** wird vom Startknoten ausgehend jeder Kindknoten besucht. Ist das Ziel noch nicht gefunden, wird im Suchbaum eine Stufe tiefer gesucht. Hierbei wird wieder jeder Kindknoten dieser Stufe besucht, bevor die nächste Stufe untersucht wird. Der Algorithmus bricht auch ab, wenn das Ziel erreicht wurde. Dieser Algorithmus kann benutzt werden, um Bäume mit

großer Tiefe zu durchsuchen. Die einzelnen Knoten sollten nicht zu viele Kindknoten haben, da ein großer Branching-Faktor zu einer exponentiellen Explosion der Knotenanzahl führt.

Die **Nichtdeterministische Suche** wählt zufällig den Knoten aus, bei dem die Kindknoten untersucht werden sollen. Die Kindknoten des gewählten Knotens werden alle nacheinander untersucht, bevor eine neue zufällige Auswahl eines Knotens erfolgt. Durch diese Eigenschaft ist es eine Mischung aus Breiten- und Tiefensuche. Dieser Algorithmus sollte benutzt werden, wenn keine Informationen über die Tiefe und Breite des Suchbaumes zur Verfügung stehen.

Heuristisch Informierte Suche

Heuristisch Informierte Suche versuchen die Suchkosten zu verringern, indem sie zusätzliche Informationen einfließen lassen, um die Auswahl des nächsten Knotens, der untersucht werden soll, zu bestimmen.

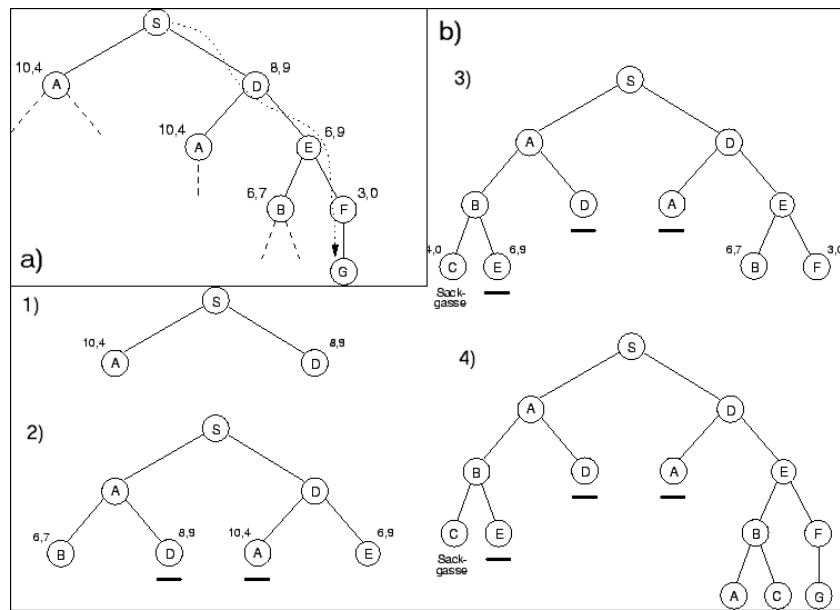


Abbildung 37: Algorithmen der Heuristisch Informierten Suche a) Hill-Climbing und b) Beam Search

Probleme, die bei der heuristischen Suche auftreten können, sind

- Vorberge - es wird nicht die globale Lösung gefunden, sondern nur eine lokale
- Platos - Kindknoten mit gleichen heuristischen Werten
- Grade - Kindknoten bieten keine Verbesserung, aber "Enkelknoten" wären eine Verbesserung

Das **Hill-Climbing** ähnelt der Tiefensuche. Für die Auswahl des Kindknotens wird aber die Heuristik benutzt, so dass der günstigste Knoten ausgesucht wird. Erreicht man während der Suche ein lokales Maximum - alle nachfolgenden Kindknoten führen zu einer Verschlechterung - wird, für eine bestimmte Anzahl von Schritten, die Suche nichtdeterministisch fortgesetzt.

Beim **Beam Search** wird derselbe Algorithmus angewendet wie bei der **Breitensuche**. Anstelle aller Kindknoten werden aber nur eine bestimmte Anzahl von Kindknoten untersucht. Die Auswahl der Kindknoten erfolgt über die Heuristik, wobei die günstigsten Knoten herausgesucht werden.

Die **Best-First Search** ähnelt dem **Hill-Climbing**. Der Unterschied besteht in der Auswahl des nächsten Knotens, wenn der Algorithmus in einer Sackgasse landet. Während beim **Hill-Climbing** solange ein Backtracking durchgeführt wird, bis ein Knoten gefunden ist, bei dem noch nicht alle Kindknoten besucht wurden, wird beim **Best-First Search**-Algorithmus der Wert der Heuristik von allen noch nicht besuchten Knoten betrachtet und aufgrund dessen der Günstigste gewählt. Tritt innerhalb der beiden Algorithmen keine Sackgasse auf, verhalten sich beide komplett identisch. Ein Problem dieses Algorithmus ist, dass nicht immer eine Lösung gefunden wird, selbst wenn eine existiert.

Optimale Netzsuche

Bei der **Optimalen Netzsuche** soll, wie der Name schon sagt, die optimale Lösung eines Suchproblems gefunden werden. Die in den vorherigen Abschnitten behandelten Algorithmen brechen immer ab, wenn irgendeine Lösung des Problems gefunden wurde. Für die Ermittlung des optimalen Weges durch den Suchbaum ist es nötig, die Kosten zu kennen, die zwischen den einzelnen Knoten existieren.

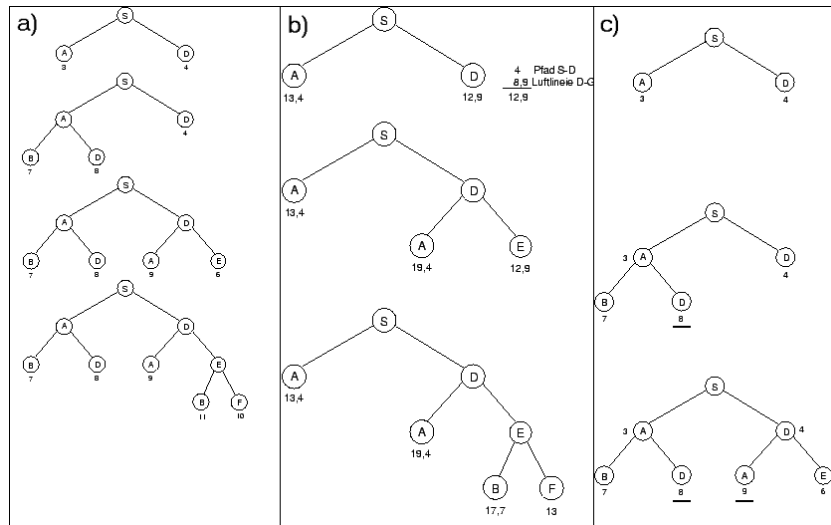


Abbildung 38: Algorithmen der Optimalen Netzsuche a) Branch-and-Bound Search, b) BaB mit Restkostenabschätzung und c) BaB mit Eliminierung von längeren Doppelwegen

Die **British Museum Procedure** eignet sich nur für kleine Graphen, da sie alle Lösungen vom Start- zum Zielknoten, mittels der Breiten- oder Tiefensuche, sucht, diese dann mit den entsprechenden Kosten bewertet und den Optimalsten auswählt.

Beim **Branch-and-Bound Search**-Algorithmus wird ausgehend vom Startknoten immer der Knoten mit den geringsten Gesamtkosten als nächster ausgewählt. Die Gesamtkosten berechnen sich dabei aus der Summe der Kosten zwischen den einzelnen Knoten, vom Startknoten bis zum betrachteten Knoten. Besonders effektiv ist der Algorithmus, wenn falsche Wege sehr hohe Kosten verursachen.

Die drei nächsten Algorithmen sind Erweiterungen der **Branch-and-Bound Search**.

Bei der **BaB mit Abschätzung der Restkosten** werden zu den Gesamtkosten die vermuteten Restkosten addiert. Je besser die Abschätzung der Restkosten, um so schneller wird der optimale Weg durch den Suchbaum gefunden. Besonders gut ist dieser Algorithmus bei der Suche nach dem optimalen Weg zwischen zwei Orten geeignet, da in diesem Fall die Luftlinien-Entfernung eine gute Abschätzung der Restkosten ist. Ein wichtiger Punkt bei der Bestimmung der Restkosten ist die Einschränkung, dass die geschätzten Kosten niemals größer sein dürfen, als die tatsächlich verbleibende Kosten.

Die **BaB mit Eliminierung längerer Doppelwege** eliminiert Wege, in welchem der Knoten schon durch einen günstigeren Weg vom Startknoten aus erreicht wurde.

Der **A*-Algorithmus** ist eine Kombination von **BaB mit Abschätzung der Restkosten** und **BaB mit Eliminierung längerer Doppelwege**.

Abbildungsverzeichnis

1.	WIESELFramework	3
2.	DIN1032-2	12
3.	Didaktikkonzept	13
4.	Semantic Web	15
5.	Schichten	15
6.	Topic Map in KnowledgeDirect	17
7.	Service-orientierte Architektur	18
8.	Eigenschaften SOA	19
9.	BELLE-Architektut	23
10.	Edna Online-Architektut	26
11.	Smete	29
12.	ECDM	32
13.	erweiterte Architektur des WIESELFramework	39
14.	WIESELfederation	40
15.	LOM - Ontologie	42
16.	OWL - Klassen	44
17.	OWL - Objekteigenschaften	45
18.	OWL - Datentypeigenschaften	45
19.	Bio-Ontologie	46
20.	LOM-Standard	47
21.	International LOM Survey	50
22.	Komponenten von WIESELFEDERATION	53
23.	Funktionsweise des Gateways	55
24.	Beispiel LOR-Ontologie	57
25.	Komposition von Werten	58
26.	Dekomposition von Werten	59
27.	Transformation zwischen Datentypen	59
28.	Abbildung von Konstanten	59
29.	Grobalgorithmus der Föderation	61
30.	SCOWrapper	68
31.	Beispiel Transformation Pipeline	70
32.	LMS Schnittstelle	76
33.	Erstellung von Knoten	77
34.	Beispiele Fachontologien	79
35.	Beispiel Sortierung	83
36.	Algorithmen Blinde Suche	93
37.	Algorithmen Heuristisch Informierte Suche	94
38.	Algorithmen Optimale Suche	96

Tabellenverzeichnis

1.	WSDL-Elemente	21
2.	Unterstützte Elemente des LOM-Standards	30
3.	Vergleich Föderationsansätze	36
4.	Erweiterung der LOM-Spezifikation	48
5.	Notwendige Elemente für die Föderation	48
6.	Wünschenswerte Elemente für die Föderation	50

Literatur

- [ADL04] ADL TECHNICAL TEAM. *Sharable Content Object Reference Model (SCORM) 2004*. zu finden unter: http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=999. Januar 2004
- [Adu] ADUNA BV. *OpenRDF.org*. zu finden unter: <http://www.openrdf.org>
- [AMC03] ARAPI, Polyxeni ; MOUMOUTZIS, Nektarios ; CHRISTODOULAKIS, Stavros: Supporting Interoperability in an Existing e-Learning Platform Using SCORM. In: *Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*, 2003
- [ANS02] AHLBORN, Benjamin ; NEJDL, Wolfgang ; SIBERSKI, Wolf: OAI-P2P: A Peer-to-Peer Network for Open Archives. In: *Proceedings for the International Conference on Parallel Processing Workshops (ICPPW'02)*, 2002. – 1530-2016/02 IEEE
- [Axi] AXIS DEVELOPMENT TEAM. *Webservice - Axis*. zu finden unter: <http://ws.apache.org/axis/>
- [BA] BARD, Jonathan ; ASHBURNER, Michael. *Open Biological Ontologies*. zu finden unter: <http://obo.sourceforge.net/>
- [BLHL01] BERNERS-LEE, Tim ; HENDLER, James ; LASSILA, Ora: The semantic web. In: *Scientific American* (2001), Mai, S. 8. – zu finden unter: <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21\&ref=sciam>
- [CAN93] CANARIE INC. *CANARIE Inc.* zu finden unter: <http://www.canarie.ca/>. 1993
- [CJ02] CHAPPELL, David A. ; JEWELL, Tyler: *Java Web Services*. 1005 Gravenstein Highway North, Sebastopol, CA 95472 : O'Reilly & Associates, Inc., 2002
- [Dev] DEVICE INDEPENDENCE WORKING GROUP. *Device Independence: Access to a Unified Web from Any Device in Any Context by Anyone*. zu finden unter: <http://www.w3.org/2001/di/>

- [din] *e-Learning: PAS 1032-1 und -2 Aus- und Weiterbildung.* zu finden unter: http://www.beuth.de/sixcms/list.php?template=bv_index_redirect_dt&query=bv_redirect_q&sv%5balias%5d=pas1032-1
- [DJ04a] DOSTAL, Wolfgang ; JECKLE, Mario: Semantik, Oden einer Service-orientierten Architektur. In: *Java Spektrum* (2004), Februar/März, S. 53–56
- [DJ04b] DOSTAL, Wolfgang ; JECKLE, Mario: Semantik und Web Services. In: *Java Spektrum* (2004), August/September, S. 58–62
- [DJK04a] DOSTAL, Wolfgang ; JECKLE, Mario ; KRIECHBAUM, Werner: Semantik und Web- Services: Beschreibung von Semantik. In: *Java Spektrum* (2004), April/Mai, S. 45–49
- [DJK04b] DOSTAL, Wolfgang ; JECKLE, Mario ; KRIECHBAUM, Werner: Semantik und Web- Services: Vokabulare und Ontologien. In: *Java Spektrum* (2004), Juni/Juli, S. 51–54
- [dub] *Dublin Core Metadata Initiative (DCMI).* zu finden unter: <http://dublincore.org/>
- [edu03] EDUCATION.AU LIMITED. *EdNA Online.* zu finden unter: <http://www.edna.edu.au/edna>. 2003
- [FMH] FRIESEN, Norm ; MCGREAL, Rory ; HUBICK, Chris. *Canadian Core Learning Resource Metadata Application Profile.* zu finden unter: <http://www.cancore.ca/indexen.html>
- [Fri04] FRIESEN, Norm: International LOM Survey: Report, 2004
- [GAP02] GRÜTZNER, Ines ; ANGKASAPUTRA, Niniek ; PFAHL, Dietmar: A systematic approach to produce small courseware modules for combined learning and knowledge management environments. In: *SEKE '02*, 2002, S. 533–539
- [Gen] GENE ONTOLOGY CONSORTIUM. *Gene Ontology Consortium.* zu finden unter: <http://www.geneontology.org/>
- [Ham03] HAMBACH, Sybille: Modularisierung von Bildungsangeboten: Problembeschreibung und Lösungsansatz. In: *Proceedings der DeLFI 2003*, 2003. – ISBN 3-88579-366-0 ISSN 1617-5468
- [Har] HARROW, Anita. *The taxonomy of Educational Objectives.* zu finden unter: <http://www.humboldt.edu/~thal/bloomtax.html>

- [HHP01] HORROCKS, Ian ; VAN HARMELEN, Frank ; PATEL-SCHNEIDER, Peter. *DAML+OIL (March 2001)*. zu finden unter: <http://www.daml.org/2001/03/daml+oil-index.html>. März 2001
- [HREW04] HATALA, Marek ; RICHARDS, Griff ; EAP, Timmy ; WILLMS, Jordan: The EduSource Communication Language: Implementing Open Network for Learning Repositories and Services. In: *2004 ACM Symposium on Applied Computing*, 2004. – ACM 1-58113-812-1/03/04
- [HSDK⁺] HOERMANN, Stefan ; SEEGER, Cornelia ; DIVAC-KRNIC, Luka ; MERKEL, Oliver ; FAATZ, Andreas ; STEINMETZ, Ralf: Building Structures of Reusable Educational Content Based on LOM
- [IEE02] IEEE LEARNING TECHNOLOGY STANDARDS COMMITTEE. *IEEE LTSC WG12: Learning Object Metadata*. zu finden unter: <http://ltsc.ieee.org/wg12/>. 2002
- [IMS] IMS GLOBAL LEARNING CONSORTIUM. *IMS Global Learning Consortium: Learning Design Specification*. zu finden unter: <http://www.msglobal.org/learningdesign/index.cfm>
- [jxt99] *JXTA technology*. zu finden unter: <http://www.jxta.org/>. 1999
- [KB04] KAISER, Siglinde ; BEHRENS, Hermann: Entwicklungsbegleitende Normung (EBN): Qualität im E-Learning. In: *Management und Qualität* (2004), September, S. 29–31
- [KS03] KARAPIPERIS, Pythagoras ; SAMPSON, Demetrios: Enhancing Educational Metadata Management Systems to support Interoperable Learning Object Repositories. In: *Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*, 2003. – 0-7695-1967-9/03 IEEE
- [LL] LEMLOUMA, Tayeb ; LAYAIDA, Nabil: Adapted Content Delivery for Different Contexts
- [LSNW02] LAGOZE, Carl ; DE SOMPEL, Herbert V. ; NELSON, Michael ; WARNER, Simeon. *The Open Archives Initiative Protocol for Metadata Harvesting*. zu finden unter: <http://www.openarchives.org/OAI/openarchivesprotocol.html>. Juni 2002
- [Mag] MAGEE, Michael: Educational Object Repositories and XML
- [MB03] MOHAN, Permanand ; BROOKS, Christopher: Learning Objects on the Semantic Web. In: *Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*, 2003. – 0-7695-1967-9/03 IEEE

- [MBD] MARTIN, David ; BURSTEIN, Mark ; DENKER, Grit. *OWL-S 1.0 Release*. zu finden unter: <http://www.daml.org/services/owl-s/1.0/>
- [Mem01] MEMBERS OF THE TOPICMAPS.ORG AUTHORIZING GROUP. *XML Topic Maps (XTM) 1.0*. zu finden unter: <http://www.topicmaps.org/xtm/1.0/>. 2001
- [mer] *MERLOT Home Page*. zu finden unter: <http://www.merlot.org/>
- [ND02] NEVEN, Filip ; DUVAL, Erik: Reusable Learning Objects: a Survey of LOM-Based Repositories. In: *Multimedia'02*, 2002. – ACM 1-58113-620-X/02/0012
- [NDS02] NEJDL, Wolfgang ; DECKER, Stefan ; SIBERSKI, Wolf. *EduTella Project Homepage*. zu finden unter: <http://edutella.jxta.org/>. 2002
- [Net01] NETERA ALLIANCE. *The BELLE Project*. zu finden unter: <http://belle.netera.ca>. 2001
- [NWQ⁺02] NEJDL, Wolfgang ; WOLF, Boris ; QU, Changtao ; DECKER, Stefan ; SINTEK, Michael ; NAEVE, Ambjörn ; NILSSON, Mikael ; PALMER, Matthias ; RISCH, Tore: EDUTELLA: A P2P Networking Infrastructure Based on RDF. In: *Proceedings of the WWW2002*, 2002. – ACM 1-58113-449-5/02/0005
- [OAS01] OASIS STANDARDS CONSORTIUM. *Advancing Web Service Discovery Standard*. zu finden unter: <http://www.uddi.org/>. 2001
- [OMR03] OLDENETTEL, Frank ; MALACHINSKI, Michael ; REIL, Dennis: Integrating Digital Libraries into Learning Environments: The LEBO-NED Approach, 2003. – 0-7695-1939-3/03 IEEE
- [ont] *Ontologie (Informatik)*. zu finden unter: <http://www.lexikon-definition.de/Ontologie-%28Informatik%29.html>
- [QN03] QU, Changtao ; NEJDL, Wolfgang: Searching SCORM Metadata in a RDF-based E-Learning P2P Network Using XQuery and Query by Example. In: *Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*, 2003
- [QN04] QU, Changtao ; NEJDL, Wolfgang: Interacting the EduTella/JX-TA Peer-to-Peer Network with Web Services. In: *Proceedings of the 2004 International Symposium on Applications and the Internet (SAINT'04)*, 2004. – 0-7695-2068-5/04 IEEE

- [Sch03] SCHULZ, Matthias: *Integration und Föderation heterogener Informationssysteme auf der Basis semantisch angereicherter Web Services*, Universität Rostock, Fachbereich Informatik, Diplomarbeit, 2003
- [See03] SEEBERG, Cornelia: *Life Long Learning: Modulare Wissensbasen für elektronische Lernumgebungen*. Springer Verlag, 2003. – ISBN 3-540-43618-9
- [SME99] SMETE OPEN FEDERATION. *Smete Digital Library*. zu finden unter: <http://www.smete.org>. 1999
- [SSS] STOJANOVIC, Ljiljana ; STAAB, Steffen ; STUDER, Rudi: eLearning based on the Semantic Web
- [suc] *Problemlösen durch Suchen*. zu finden unter: <http://www.iicm.edu/greif/node6.html>
- [Sun] SUN MICROSYSTEMS, INC. *Java Technologie*. zu finden unter: <http://java.sun.com>
- [Swe04] SWERTZ, Christian: *Didaktisches Design*. Bielefeld : W. Bertelsmann Verlag, Fachverlag für Bildung und Beruf, 2004. – ISBN 3-7639-0193-0
- [TDV] TERNIER, Stefaan ; DUVAL, Erik ; VANDEPITTE, Pieter: LOMster: Peer-to-peer Learning Object Metadata
- [TFOM] TAMURA, Yasuhisa ; FRIESEN, Norm ; OKAMOTO, Toshio ; MCGREAL, Rory: Metadata for Synchronous and Asynchronous Collaborative Learning Environments
- [The] THE APACHE SOFTWARE FOUNDATION. *The Jakarta Site - Apache Jakarta Tomcat*. zu finden unter: <http://jakarta.apache.org/tomcat/>
- [TZ] TERGAN, Sigmar-Olaf ; ZENTEL, Peter: Lernplattformen und die Zukunft des E-Learning
- [web] *eduplone - Grundlagen der Webdidaktik*. zu finden unter: <http://eduplone.net/concepts/webdidaktik/Theoretische-Grundlagen/de>
- [Wie92] WIEDERHOLD, Gio: Mediators in the Architecture of Future Information Systems, 1992. – 0018-9162/92/0300-0038 IEEE
- [Wor97] WORLD WIDE WEB CONSORTIUM W3C. *Resource Description Framework(RDF) / W3C Semantic Web Activity*. zu finden unter: <http://www.w3.org/RDF/>. 1997

- [Wor01a] WORLD WIDE WEB CONSORTIUM W3C. *W3C Semantic Web*. zu finden unter: <http://www.w3.org/2001/sw/>. 2001
- [Wor01b] WORLD WIDE WEB CONSORTIUM W3C. *Web Services Description Language (WSDL)*. zu finden unter: <http://www.w3.org/TR/wsdl>. 2001
- [Wor03] WORLD WIDE WEB CONSORTIUM W3C. *Simple Object Access Protocol (SOAP) 1.1*. zu finden unter: <http://www.w3.org/TR/soap/>. 2003
- [Wor04a] WORLD WIDE WEB CONSORTIUM W3C. *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0*. zu finden unter: <http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/>. Januar 2004
- [Wor04b] WORLD WIDE WEB CONSORTIUM W3C. *Extensible Markup Language (XML) 1.0 (Third Edition)*. zu finden unter: <http://www.w3.org/TR/REC-xml/>. Februar 2004
- [Wor04c] WORLD WIDE WEB CONSORTIUM W3C. *OWL Web Ontology Language*. zu finden unter: <http://www.w3.org/TR/owl-features/>. Februar 2004
- [Wor04d] WORLD WIDE WEB CONSORTIUM W3C. *RDF Vocabulary Description Language 1.0: RDF Schema*. zu finden unter: <http://www.w3.org/TR/rdf-schema/>. Februar 2004
- [YT03] YANG, Jin-Tan D. ; TSAI, Chun-Yen: An Implementation of SCORM-compliant Learning Content Management System - Content Repository Management System. In: *Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*, 2003. – 0-7695-1967-9/03 IEEE
- [ZGD03] ZGDV E.V. ROSTOCK. *KnowledgeDirect: Wissensmanagement in thematischen Netzwerken*. zu finden unter: <http://www.knowledge-direct.de/>. 2003

Eidesstattliche Versicherung

Ich versichere eidesstattlich durch eigenhändige Unterschrift, dass ich die Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, habe ich als solche kenntlich gemacht. Ich weiß, dass bei der Abgabe einer falschen Versicherung die Prüfung als nicht bestanden zu gelten hat.

Rostock, 21. Oktober 2004

EINVERSTÄNDNISERKLÄRUNG

Ich erkläre mich damit einverstanden, dass ein Exemplar meiner Diplomarbeit in der Universitätsbibliothek der Universität Rostock aufbewahrt und für die allgemeine Nutzung zugänglich gemacht wird. Ich nehme zur Kenntnis, dass dies die Wirkung einer Veröffentlichung im urheberrechtlichen Sinne hat.

Rostock, 21. Oktober 2004